PERANCANGAN DAN IMPLEMENTASI MANAJEMEN BANDWITH SIMPLE QUEUE DAN QUEUE TREE MENGGUNAKAN FIREWALL PROTOCOL LAYER 7
(STUDI KASUS iTCentrum)

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat

Untuk memperoleh Gelar Sarjana

Jurusan Teknik Informatika

Disusun oleh :

Nama : SUHARIANTO

NIM : 08523150

JURUSAN TEKNIK INFORMATIKA

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITAS ISLAM INDONESIA

YOGYAKARTA

2013

HALAMAN JUDUL

PERANCANGAN DAN IMPLEMENTASI MANAJEMEN BANDWITH SIMPLE QUEUE DAN QUEUE TREE MENGGUNAKAN FIREWALL PROTOCOL LAYER 7 MIKROTIK
(STUDI KASUS iTCentrum)

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat

Untuk Memperoleh Gelar Sarjana

Jurusan Teknik Informatika

Disusun oleh :

Nama : SUHARIANTO

NIM : 08523150

JURUSAN TEKNIK INFORMATIKA

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITAS ISLAM INDONESIA

YOGYAKARTA

2013

LEMBAR PENGESAHAN DOSEN PEMBIMBING

PERANCANGAN DAN IMPLEMENTASI MANAJEMEN BANDWITH SIMPLE QUEUE DAN QUEUE TREE MENGGUNAKAN FIREWALL PROTOCOL LAYER 7 MIKROTIK
(STUDI KASUS iTCentrum)

TUGAS AKHIR

oleh :

Nama : SUHARIANTO

NIM : 08523150

Yogyakarta, 17 Oktober 2013

Pembimbing,

Ahmad M. Rafie Pratama, ST, M.IT.

LEMBAR PENGESAHAN PENGUJI

PERANCANGAN DAN IMPLEMENTASI MANAJEMEN BANDWITH SIMPLE QUEUE DAN QUEUE TREE MENGGUNAKAN FIREWALL PROTOCOL LAYER 7 MIKROTIK TUGAS AKHIR

Disusun Oleh :

Nama : Suharianto

No. Mahasiswa : 08523150

Telah Dipertahankan di Depan Sidang Penguji Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana Jurusan Teknik Informatika Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 08 November 2013

Tim Penguji,

Ahmad M. RafiePratama, S.T., MIT.
Ketua

Ari Sujarwo, S.Kom., MIT.
Anggota I

Ahmad Luthfi, S.Kom., M.Kom.
Anggota II

Mengetahui,
Ketua Jurusan Teknik Informatika
Fakultas Teknologi Industri Universitas Islam Indonesia

Yudi Prayudi, S.Si., M.Kom.

LEMBAR PERNYATAAN KEASLIAN
HASIL TUGAS AKHIR

Saya yang bertanda tangan di bawah ini :

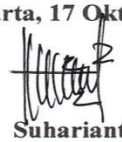
Nama : **Suharianto**

NIM : **08523150**

Menyatakan bahwa keseluruhan komponen isi dalam penulisan laporan tugas akhir ini adalah hasil dari karya saya sendiri. Apabila dikemudian hari terdapat dan terbukti bahwa ada beberapa bagian dari karya yang bukan hasil karya saya sendiri, maka saya siap menanggung segala resiko dan konsekuensinya.

Demikian pernyataan ini saya buat tanpa ada paksaan dan tekanan dari manapun, dan semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 17 Oktober 2013



Suharianto

HALAMAN PERSEMBAHAN

Untuk kedua orang tua kami yang telah mendidik dan membiayai kami sehingga kami bisa melanjutkan studi kami hingga universitas seperti sekarang ini.

Untuk para dosen yang telah memberikan kami semua ilmu yang berguna di dalam maupun di luar perkuliahan

Untuk teman-teman kami yang telah memberikan support, bantuan moril, bantuan ilmu, dan banyak lagi.

Untuk almamater kami yang menjadi rumah belajar kami selama masa studi di sini.

HALAMAN MOTTO

I'VE NO FAILED. I'VE JUST FOUND 10.000 WAYS THAT WON'T WORK
(Thomas a. Edison)

BELIEVE IN YOUR SELF, THE RESULTS WILL SPEAK FOR YOU
(JUAN ZAMBRANO)

SIBUK UNTUK HIDUP ATAU SIBUK UNTUK MATI
(ANDY DUFRENSE THE SHAWSHANK REDEMPTION)

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Puji syukur saya panjatkan kepada Allah SWT Tuhan Semesta Alam atas rahmat dan karunia-Nya sehingga saya dapat menyelesaikan skripsi ini. Shalawat serta salam saya ucapkan pula kepada teladan kami, Rasulullah SAW.

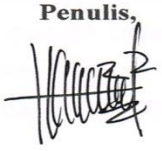
Ucapan terima kasih saya haturkan kepada semua pihak yang telah membantu dan memberikan dukungan selama penyusunan tugas akhir ini diantaranya :

1. Kedua orang tua, atas kasih saying dan doa tulus yang tak pernah lelah diucapkan serta dukungan yang telah menguatkan.
2. Adik-adik saya yang selalu memberikan semangat.
3. Bapak Ir. Gumbolo Hadi Susanto, M.Sc. selaku Dekan Fakultas Tenologi Industri Universitas Islam Indonesia.
4. Bapak Yudi Prayudi, S.Si., M.Kom. selaku Ketua Program Studi Teknik Informatika Universitas Islam Indonesia.
5. Bapak Ahmad M. Rafie Pratama, S.T., MIT. selaku dosen pembimbing yang selalu memberi motivasi dan inspirasi yang sangat berharga bagi penulis.
6. Seluruh Bapak dan Ibu dosen Teknik Informatika Universitas Islam Indonesia untuk ilmu yang telah dilimpahkan kepadaku.
7. Bapak Hamid, S.T., M.Eng. untuk dukungan ilmu dan semangatnya untuk menyelesaikan tugas akhir ini.
8. Ivana Yuristira, untuk semua dukungan, bantuan, semangat yang diberikan.
9. Citra Affarudin dan Yuda, untuk semua ilmu yang diajarkan.
10. Bagus ogud, Riyan Katro, azizi asipoy, Indwa Gabon, Danan Kibo, Dwi Mamat, Bino Bienz, Selsy, Pima, Gina dan saheb – saheb lainnya atas saran – saran dan bantuannya.
11. Teman-teman Kelas B yang telah mendukung saya.
12. SNIPER UII 08 yang selalu mendukung dan mengajarkan segala hal.

Penulis menyadari bahwa skripsi ini masih kurang sempurna, dan untuk itulah penulis mengharapkan saran dan kritik yang membangun demi penyempurnaan skripsi ini. Semoga skripsi ini bermanfaat bagi pembaca, khususnya bagi pengembangan manajemen bandwith dan implementasi internet sehat serta dapat dijadikan referensi dalam menyusun kebijakan yang berhubungan dengan perkembangan Universitas Islam Indonesia di Yogyakarta dan menjadi suatu karya yang bisa memberi dampak positif.

Yogyakarta, 17 Oktober 2013

Penulis,



Suharianto

SARI

Dengan semakin meningkatnya pertumbuhan dan penggunaan jaringan Internet dewasa ini, tentu akan mempengaruhi ketersediaan bandwith dan keamanan dalam suatu jaringan. Penggunaan manajemen bandwith dan pengunaan firewall memberikan manfaat yang sangat baik untuk iTCentrum yang sekaligus merupakan menjadi pusat pelatihan bersertifikasi bagi mahasiswa. Oleh karena itu, perlu diketahui faktor-faktor apa yang mempengaruhi penggunaan bandwith yang tidak sesuai.

Metode penelitian dalam penelitian ini menggunakan studi pustaka dan konfigurasi sistem serta mengimplementasikan beberapa skenario konfigurasi yang mendukung manajemen bandwith serta penggunaan firewall sebagai pengamanan utama dengan dasar file ekstensi sebagai bahan dari data yang akan diambil. Variabel yang diteliti yaitu *speed-limit* dan *drop paket*, *quality of service* dan *resource memory router* yaitu besar beban memori terhadap router.

Dari hasil perancangan dan implementasi manajemen bandwith *queue tree* memeberikan hasil yang lebih baik dibandingkan dengan *simple queue*. Hal ini dapat dilihat dari data *troughput* pada *queue tree* lebih besar dengan selisih 16.00-212.00 kbps untuk *throughput* 1, 15.00-203.00 kbps untuk *throughput* 2, dan 15.00-210.00 kbps untuk *throughput* 3, persentase *packet loss* pada *queue tree* lebih sedikit dibandingkan *simple queue* dengan nilai sebesar 0-15% packet loss 1, 0-86% untuk packet loss 2, 0-42% untuk packet loss 3, dan beban *memory* untuk *simple queue* memberikan beban *Max* sebesar 19.05 MB, *Average* 18.41 MB, dan *Current* 19.05 MB sedangkan *queue tree* dengan penambahan firewall *protocol layer 7* memberikan beban *Max* 18.74 MB, *Average* 18.29 MB, dan *Current* sebesar 17.83 MB dimana queue tree lebih ringan dibandingkan dengan simple queue. Selain kualitas manajemen bandwith *queue tree* yang lebih baik penggunaan firewall pada *queue tree* juga dapat memberikan nilai keamanan yang lebih baik.

Kata kunci : MikroTik, Manajemen Bandwith, Firewall, Protocol Layer 7

TAKARIR

Queue	:	Antrian
Simple Queue	:	Limit berdasarkan date rate
Queue Tree	:	Limit berdasarkan protocol, ports, ip address
Firewall	:	Tembo kapi
Filter Rule	:	Aturan Penyaringan paket
Firewall Mangle	:	Metode menandai paket
Protocol Layer 7	:	Mencocokan dan mencari pola yang sesuai
Throughput	:	Bandwith actual pada suatu ukuran tertentu
Packet Loss	:	Paket yang hilang
Delay	:	Waktu yang dibutuhkan untuk pengiriman paket
Speed-limit	:	Besar kecepatan bandwith
Action Accept	:	Aksi menerima paket
Action Drop	:	Aksi membuang paket
Client	:	Klien
Matcher	:	Mencocokan
Request	:	Permintaan
Replay	:	Mengulang
Collision	:	Bentrokan
Packet	:	Paket
Remote	:	Alat Pengendali
Traffic Flow	:	Aliran Lalu-lintas
Rule	:	Aturan
Policies	:	Kebijakan

Computer Network	:	Komputer yang saling terhubung sehingga dapat berbagi sumber

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I
PENDAHULUAN

1.1 Latar Belakang

Computer network atau jaringan komputer ialah dua atau lebih komputer saling terhubung sehingga dapat berbagi sumber daya (*resources*). Dengan begitu diharapkan dapat memunculkan efensiasi, sentralisasi, dan opstimasi kerja. Seiring dengan pesatnya kebutuhan akan informasi-informasi yang ada dalam internet, dituntut cara-cara dalam memenuhi kebutuhan tersebut. MikroTik RouterOs adalah sistem operasi dan perangkat lunak yang dapat digunakan untuk menjadikan komputer menjadi *router network* yang handal, mencakup berbagai fitur yang dibuat untuk *ip network* dan jaringan *wireless*, cocok digunakan oleh *Internet Service Provider* dan *provider hotspot*. Namun, bagaimana kalau jaringan yang ada tidak sesuai dengan kecepatan maksimal yang diinginkan. Banyak hal yang sangat mempengaruhi kecepatan dari suatu jaringan internet yang ada pada tempat tersebut. Hal ini tentu sangat dipengaruhi oleh besarnya bandwith yang diberikan terhadap suatu server jaringan internet. Bandwith (*data transfer rate*) adalah suatu data keluar masuk (pengunduhan atau pengunggahan) dalam suatu jangka tertentu.

Penggunaan bandwith yang tidak efektif pada sebuah jaringan dapat memicu jaringan internet dari sebuah tempat menjadi sangat lambat. Hal ini dapat disebabkan karena kurang maksimalnya pemanfaatan bandwith dari *server* ke *client-client* yang ada dibawahnya. *client* mendapatkan kapasitas bandwith yang berbeda-beda, *client* yang mendapatkan kapasitas bandwith yang besar dapat menyebabkan jaringan internet client yang lain menjadi lambat. Pengunaan bandwith yang tidak sesuai juga dapat menyebabkan suatu *down* terhadap jaringan itu sendiri. Proses pengunduan yang berlebihan merupakan salah satu pemicu utama dari lambatnya suatu jaringan. Oleh karena itu, solusi yang sangat baik yaitu bagaimana mengatur bandwith jaringan itu sendiri sehingga bandwith yang ada dapat dimaksimalkan lebih optimal untuk

penggunaannya. *Quality of service* atau manajemen bandwith merupakan salah satu solusi yang dapat dilakukan.

Dalam perkembangannya MikroTik juga ternyata mampu menemukan cara-cara untuk melakukan manajemen bandwith secara optimal untuk memaksimalkan kinerjanya. Beberapa langkah dapat diterapkan dalam proses manajemen bandwith. Beberapa contoh yang dapat digunakan yaitu *queue* dan *protokol layer 7* agar dapat dilakukan proses perbandingan terhadap manajamen bandwith yang lebih baik.

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas maka didapat suatu rumusan masalah yaitu bagaimana sebenarnya fingsionalitas dan kinerja manajemen bandwith serta meningkatkan keamanan jaringan dari rancanangan sistem *simple queue* dan *queue tree* menggunakan *firewall Layer 7 Protocol*.

1.3 Batasan Masalah

Adapun batasan masalah yang terdapat dalam penelitian ini diantaranya :

a. Mikrotik Routerboard 493 digunakan sebagai router dan bandwith manajemen.
b. Variabel yang akan diamati yaitu *drop packet* dan *speed-limit bandwith*, *Throughput*, *Delay*, *Packet Loss* serta penggunaan *resource memory* yang diimplementasikan pada manajemen bandwith menggunakan *protokol layer 7 queue tree* dam *simple queue*.
c. Menganalisa hasil rancangan sistem dari kinerja *queue tree* dengan *protokol layer 7* dan *simple queue* dengan melakukan pengujian kinerja dan fungsionalitas berdasarkan penggunaan bandwith, memori router, serta *filtering content*.

1.4 Tujuan Penelitian

Tujuan penelitian dari tugas akhir ini adalah untuk mengetahui kinerja queue tree dengan menggunakan *protokol layer 7* dan *simple queue* pada saat melakukan

filtering terhadap suatu *content* dan seberapa maksimal fungsi manajemen bandwith yang ada dalam *queue tree* dengan *protokol layer 7* dan *simple queue*.

1.5 Manfaat Penelitian

Melalui hasil penelitian dan pengimplementasian rancangan sistem dari manajemen bandwith dalam penggunaan queue tree dengan *firewall protokol layer 7* dan *simple queue* ini, diharapkan pihak-pihak yang ingin menggunakan maupun ingin mempelajari manajemen bandwith menggunakan *queue tree* dengan *firewall protokol layer 7* maupun *simple queue* yang ada pada MikroTik mendapatkan gambaran tentang fungsi serta kinerja dari masing-masing parameter tersebut serta juga dapat meningkatkan keamanan jaringan yang ada di iTCentrum..

1.6 Metodologi Penelitian

Pada tugas akhir ini, penulis menggunakan beberapa metodologi penelitian yang digunakan sebaga acuan dalam mengerjakan tugas akhir ini, antara lain :

1.6.1 Studi Pustaka

Studi pustaka digunakan untuk mendapatkan dan menghasilkan informasi tambahan yang digunakan sebagai acuan dan berhubungan erat dengan penelitian melalui buku, media cetak, internet, maupun nara sumber yang berhubungan dengan penelitian.

1.6.2 Implementasi dan Analisis Sistem

Adapun mengenai langkah pengerjaan, penulis menyusun langkah pengerjaan analisis perbadingan kinerja *protokol layer 7* dan *simple queue* berdasarkan perolehan dari studi pustaka, antara lain :

a. Analisis Kebutuhan Sistem

   Tahapan awal yang dilakukan yaitu melakukan analisis dan identifikasi mengenai topologi jaringan iTCentrum, perangkat lunak, router yang digunakan, serta hal-hal apa saja yang dibutuhkan dalam penelitian

mengenai analisis perbandingan manajemen bandwith menggunakan *protokol layer 7* dan *simple queue*.

b. Pengadaan Perangkat Keras

Tahan ini merupakan tahap pengadaan perangkat keras yang akan digunakan seperti mikrotik routerboard 493 yang digunakan sebagai konfigurasi bandwith manajemen.

c. Pembangunan Sistem

Pada tahap ini merupakan tahapan instalasi dan konfigurasi routerboard, aplikasi, konfigurasi *queue tree* dengan *protokol layer 7*, konfigurasi *simple queue*, serta pemberian konfigurasi-konfigurasi lain yang mendukung dalam tahapan pembangunan sistem ini.

d. Metode Analisis

Tahap ini penulis menentukan statistik yang digunakan untuk menganalisa, menentukan pengujian, dan skenario simulasi yang akan digunakan.

e. Pengujian Analisis

Setelah semua keperluan dan konfigurasi dibuat, penulis kan menjalankan simulasi dan melihat apakah ada proses yang tidak sesuai maupun *error* yang tidak sesuai dengan skenario dan konfigurasi yang telah dibuat.

f. Pengambilan Kesimpulan

Tahap terakhir yaitu melakukan pengumpulan analisa pada statistik data, yang kemudian hasil dari anilisis tersebut dapat ditarik kesimpulan konfigurasi mana yang memiliki kinerja manajemen bandwith paling optimal, dan apakah *firewall protokol layer 7* dapat *filtering content* yang baik.

1.7 Sistematika Penulisan

BAB I Pendahuluan

Membahas masalah latar belakang masalah, batasan masalah, rumusan masalah, tujuan penelitian, manfaat penelitian, dan metodologi yang

diangkat menjadi materi laporan tugas akhir Perancangan dan Implementasi Manajemen Bandwith Simple Queue dan Queue Tree Menggunakan Protocol Layer 7 Dengan Simple Queue Pada Mikrotik Studi Kasus iTCentrum.

BAB II Landasan Teori

Membahas uraian dasar-dasar teori yang digunakan dalam perancangan dan pembangunan sebuah manajemen bandwith menggunakan *protokol layer 7* dan *simple queue*.

BAB III Metodologi

Membahas uraian tentang gambaran umum sistem, analisis kebutuhan perangkat keras dan perangkat lunak, metode analisis yang akan dipakai serta pembangunan sistem yang akan digunakan.

BAB IV Hasil dan Pembahasan

Pada bagian ini memuat tentang dokumentasi pengujian sistem yaitu dengan menganalisis terhadap manajeman bandwith dan kinerja *protokol layer 7* dan *simple queue* yang telah dibuat yang kemudian ditarik kesimpulan dari data yang didapat.

BAB V Penutup

Memuat kesimpulan-kesimpulan dari seluruh rangkaian proses dan hasil perancangan sistem, implementasi sistem, dan proses analisis yang didapatkan berdasarkan hasil penelitian studi komparasi manajemen bandwith menggunakan *protokol layer 7* pada *queue tree* dan *simple queue* pada mikrotik.

BAB II
LANDASAN TEORI

2.1 Jaringan Komputer

Computer Network atau jaringan komputer adalah sekelompok komputer otonom yang saling menggunakan protokol komunikasi melalui media komunikasi (Dharma Oetomo (1), 2003, hal 7) sehingga dapat berbagi data, informasi, program aplikasi, dan perangkat keras seperti *printer*, *scanner*, *CD-Drive* ataupun harddisk, serta memungkinkan untuk saling berkomunikasi secara elektronik.

Adapun sejumlah potensi jaringan komputer, antara lain :

1. Mengintegrasikan dan berbagi pakai peralatan

   Jaringan komputer memungkinkan pengguna bersama peratalan komputer berbagai merek, yang semula tersebar di berbagai ruangan, unit, dan departemen sehingga meningkatkan efektifitas dari pengguna sumber daya tersebut.

2. Komunikasi

   Jaringan komputer memungkinkan terjadinya komunikasi antar pemakai komputer. Selain itu tersedia aplikasi *teleconference* yang memungkinkan dilakukan rapat atau pertemuan tanpa harus meninggalkan kerja.

3. Mengintegrasikan data

   Jaringan komputer diperlukan untuk mengintegrasikan data antar komputer-komputer *client* sehingga dapat diperoleh suatu data yang relevan.

4. Perlindungan data dan informasi

   Jaringan komputer memudahkan upaya perlindungan data yang terpusat pada server, melalui pengaturan hak akses dari para pemakai serta penerapan sistem password.

5. Sistem terdistribusi

   Jaringan komputer dimanfaatkan pula untuk mendistribusikan proses dan aplikasi sehingga dapat mengurangi terjadinya *bottleneck* atau tumpukan pekerjaan pada suatu bagian.

6. Keteraturan aliran informasi

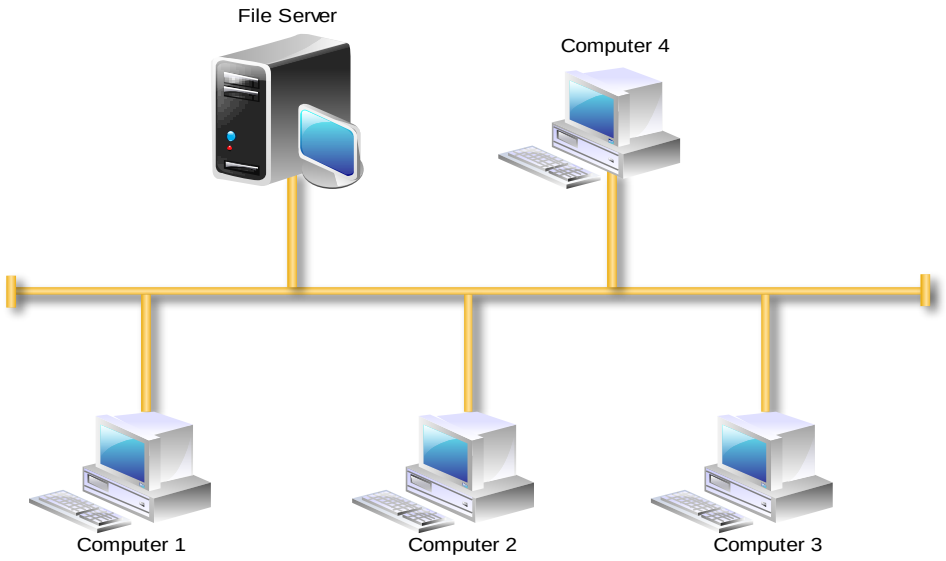
   Jaringan komputer mampu mengalirkan data-data komputer *client* dengan cepat untuk diintegrasikan dalam komputer server. Selain itu, jaringan mampu untuk mendistribusikan infromasi secara kontinu kepada pihak-pihak terkait yang membutuhkannya.

2.2 Topologi Jaringan

Topologi atau arsitektur jaringan merupakan pola hubungan antar terminal dalam suatu sistem jaringan komputer. Topologi ini akan mempengaruhi tingkat efektifitas kinerja jaringan. Ada beberapa jenis topologi yang dapat diimplementasikan dalam jaringan. Namun, bentuk topologi yang utama adalah topologi bus, topologi ring, dan topologi star.

2.2.1 Topologi Bus

Topologi bus merukapan topologi yang menghubungkan semua terminal ke satu jalur komunikasi yang kedua ujungnya ditutup dengan terminator. Terminator adalah perangkat yang menyediakan resistansi listrik untuk menyerap sinyal pada akhir transmisi sambungan agar sinyal tidak terlontar kembali dan diterima lagi oleh stasiun jaringan.



Gambar 2.1 Topologi Jaringan Bus

Keuntungan :

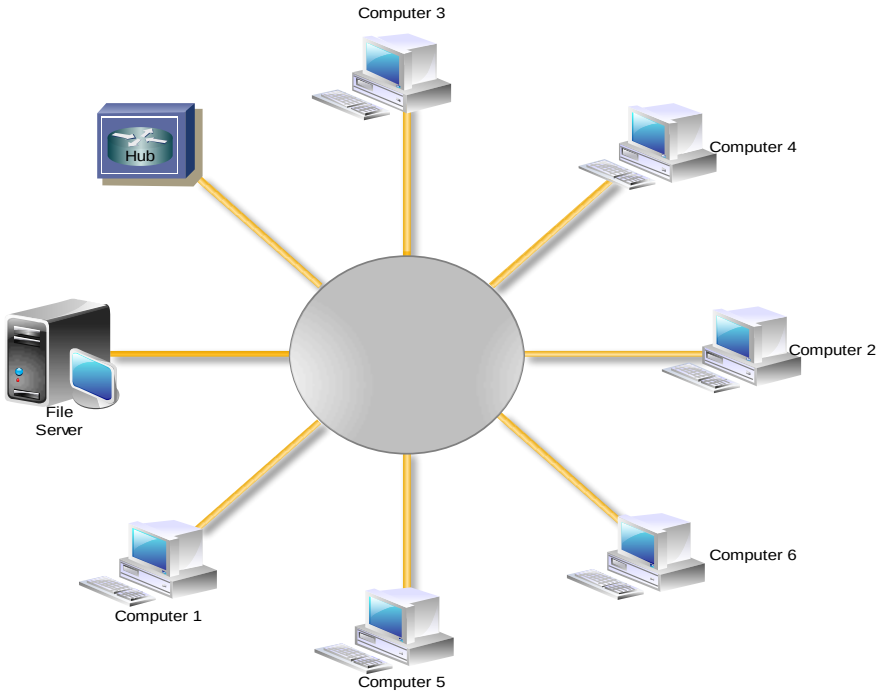
1. Murah, karena tidak memakai banyak media dan kabel yang dipakai sudah umum.
2. Setiap komputer dapat saling berhubungan langsung.

Kerugian :

Sering terjadi *hang* (*crasstalk*) ketika lebih dari satu pasang memakai jalur di waktu yang sama.

2.2.2 Topologi Ring

Pola dari topologi ring hampir sama dengan topologi bus, tetapi kedua terminal yang berada di ujung saling dihubungkan sehingga hubungan antar terminal berlangsung dalam suatu lingkaran tertutup.



Gambar 2.2 Topologi Jaringan Ring

Keuntungan :

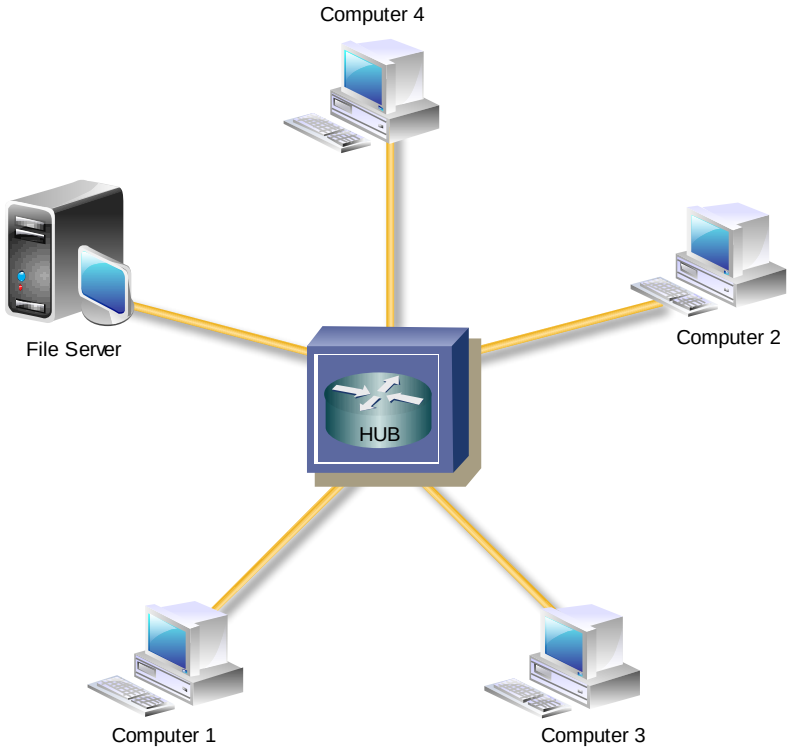
1. Kegagalan koneksi akibat gangguan media dapat diatasi dengan jalur lain yang masih terhubung.
2. Penggunaan sambungan *point to point* membuat *transmission error* dapat diperkecil.

Kerugian :

Transfer data menjadi lambat bila data yang dikirim melalui banyak komputer.

2.2.3 Topologi Star

Pada topologi star, terdapat sebuah terminal pusat (hub/switch) yang mengatur dan mengendalikan semua kegiatan komunikasi data. Trafik data mengalir dari *node* ke terminal pusat dan diteruskan ke *node* (*station*) tujuan.



Gambar 2.3 Topologi Jaringan Star

Keuntungan :

1. Akses ke *station* lain (*client* atau server) cepat.
2. Dapat menerima workstation baru selama port di *central node* (hub/switch) tersedia
3. Hub/switch bertidak sebagai konsentrator.
4. Hub/switch dapat disusun seri (bertingkat) untuk menambah jumlah *station* yang terkoneksi dijaringan.
5. Mendukung user yang banyak disbanding topologi bus, maupun ring.

Kerugian :

Bila *traffic* data cukup tinggi dan terjadi *collision*, semua komunikasi akan ditunda, dan koneksi akan dilanjutkan dengan cara random ketika hub/switch mendeteksi tidak ada jalur yang sedang digunakan oleh node lain.

2.3 Sejarah MikroTik

Cisco tentunya bukan nama yang asing lagi dalam dunia router, yaitu perangkat yang berfungsi untuk mengerahkan alamat diinternet. Namun, selain Cisco terdapat nama lain yang dikenal sebagai salah satu solusi murah untuk membangun sebuah router, yaitu MikroTik RouterOS™. MikrtoTik RouterOS™ adalah sistem operasi yang dirancang khusus untuk *network router*.Dengan sistem operasi ini, dapat dibuat router dari komputer rumahan (PC).

MikroTik adalah perusahaan kecil yang berpusat di Latvia, berseblahan dengan Rusia. Pembentukannya diprakarsai oleh John Trully dan Arnis Riekstins. Prinsip dasar mereka bukan membuat Wireless ISP (WISP), tetapi membuat program router yang handal dan dapat dijalankan diseluruh dunia.Untuk Negara berkembang, solusi MikrtoTik sangat membantu ISP atau perusahaan-perusahaan kecil yang ingin bergabung dengan internet. Walaupun sudah banyak tersedia perangkat router mini sejenis NAT, MikroTik merupakan solusi terbaik dalam beberapa kondisi penggunaan komputer dan perangkat lunak.

2.3.1 MikroTik RouterOS™

MikroTik merupakan bentuk perangkat lunak yang dapat diinstall pada komputer rumahan (PC) melalui CD. File image MikroTik RouterOS dapat diunduh dari website resmi Mikrotik, www.mikrotik.com. Namun, File image ini merupakan versi *trial* MikroTik yang hanya dapat digunakan dalam waktu 24 jam saja. Untuk dapat menggunakannya secara *full time* harus membeli lisensi *key* dengan catatan satu lisensi *key* hanya untuk satu harddisk.

2.3.2 Fitur-fitur MikroTik

1. ***Addresslist***

   Pengelempokan IP address berdasarkan nama.

2. ***Asynchronous***

Mendukung serial PPP dial-in/dial-out, dengan otentikasi CHAP, PAP, MSCHAPv1, dan MSCHAPv2, radius, dial on deman, modem pool hingga 128 ports.

3. ***Bonding***

   Mendukung dalam pengkombinasian beberapa antar muka Ethernet ke dalam 1 pipa koneksi yang cepat.

4. ***Bridge***

   Mendukung fungsi bridge spinning tree, multiple bridge interface, bridge firewalling.

5. ***Data Rate Management***

   QoS berbasi HTB dengan penggunaan burst, PCQ, RED, SFQ, FIFO queue, CIR, MIR, limit antar peer to peer.

6. ***DHCP***

   Mendukung DHCP tiap antarmuka; DHCP relay; DCHP Client; multiple network DHCP; static dan dynamic DHCP leases.

7. ***Firewall* dan *NAT***

   Mendukung pemfilteran koneksi peer to peer, source NAT dan destination NAT. Mampu memfilter berdasarkan MAC, IP address, range port, protokol IP, pemilihan opsi protokol seperti ICMP, TCP flags dan MSS.

8. ***Hotspot***

   Hotspot gateway dengan otentikasi RADIUS. Mendukung limit data rate, SSL, HTTPS.

9. ***IPSec***

   Protokol AH dan ESP untuk IPSec; MODP Diffie-Hellman groups 1, 2, 5; MD5 dan algoritma SHA1 hashing; algoritma enkripsi menggunakan DES, 3DES, AES-128, AES-192, AES-256; Perfect Forwarding Secresy (PFS) MODP groups 1, 2, 5.

10. ***ISDN***

    Mendukung ISDN dial-in / dial-out dengan otentikasi PAP, CHAP, MSCHAPv1, dan MSCHAPv2 serta Radius. Mendukung 128K bundle, Cisco HDLC, x751, x751ui, x751bui line protokol.

11. ***M3P***

    MikroTik Protokol Packet Packer untuk wireless links dan Ethernet.

12. ***MNDP***

    MikroTik Discovery Neighbor Protocol, juga mendkung Cisco Discovery Protocol (CDP).

13. ***Monitoring / Accounting***

    Laporan traffic IP, log, statistik graphs yang dapat diakses melalui HTTP.

14. ***NTP***

    Network Time Protocol untuk server dan client; sinkronisasi menggunakan system GPS.

15. ***Pointto Point Tunneling Protocol***

    PPTP, PPPoE, dan L2TP Access Concentrators; Protokol otentikasi menggunakan PAP, CHAP, MSCHAPv1, MSCHAPv2; otentikasi dan laporan RADIUS; enkripsi MPPE; kompresi untuk PPOE; Limit data rate.

16. ***Proxy***

    Cache untuk FTP dan HTTP proxy server; HTTPS proxy; transparent proxy untuk DNS dan HTTP; mendukung protokol SOKCS; mendukung parent proxy; Statik DNS.

17. ***Routing***

    Routing statik dan dinamik; RIP v1 / v2, OSPF v2, BGP v4.

18. ***SDSL***

    Mendukung Single Line DSL; mode pemutusan jalur koneksi dan jaringan.

19. ***Simple Tunnels***

    Tunnel IPIP dan EoIP (Ethernet over IP).

20. ***SNMP***

    Mode akses read-only

21. ***Synchronous***

    V.35, V.24, E1 / T1, X21, DS3 (T3) media types; sync-PPP, Cisco, HDLC; Frame Relay Line Protocol; ANSI-617d (ANDI atau annex D) dan Q933a (CGITT atau annex A); Frame Relay jenis LMI

22. ***Tool***

    Ping; traceroute; bandwith test; ping flood; telnet; SSH; packet sniffer; Dinamik DNS update.

23. ***UPnP***

    Mendukung antarmuka universal Plug dan Play

24. ***VLAN***

    Mendukung Virtual LAN IEEE802.1q untuk jaringan Ethernet dan Wireless; multiple VLAN; VLAN bridging.

25. ***VOIP***

    Mendukung aplikasi voice over IP.

26. ***VRRP***

    Mendukung Virtual Router Redudant Protocol

27. ***WinBox***

    Aplikasi mode GUI untuk meremote dan mengonfigurasi MikroTik RouterOS.

2.4 *Quality* of *Service*

Dengan mengelola jaringan, sangat penting untuk mengendalikan pemakaian bandwith yang akan digunakan oleh komputer user. Jika hal ini tidak dikendalikan, maka akan terjadi pemakaian bandwith secara berlebihan oleh satu atau beberapa user. Pemakaian yang berlebihan tersebut akan menyebabkan komputer user yang lain tidak lagi mendapatkan alokasi bandwith. Pada akhirnya jaringan yang ada tidak lagi memberikan layanan (*service*) secara maksimal kepada seluruh user yang ada. Keadaaan ini akan bertambah parah jika ternyata jaringan yang ada hanya memiliki alokasi bandwith internet yang terbatas dan pengguna jaringan rakus akan aktifitas pengunduhan file-file besar dari internet.

Router mikrotik memiliki fitur *Queue* yang dapat melakukan pengaturan (manajemen) alokasi bandwith bagi setiap komputer user. Dengan menerapkan manajemen bandwith, dapat melakukan perbaikan terhadap kualitas layanan di jaringan (*Quality of Service*). *Quality of Service* atau QoS akan memberikan jaminan alokasi bandwith minimum pada setiap komputer user di dalam jaringan, sehingga setiap komputer user tidak perlu khawatir akan tidak kebagian bandwith.

2.4.1 Simple Queue

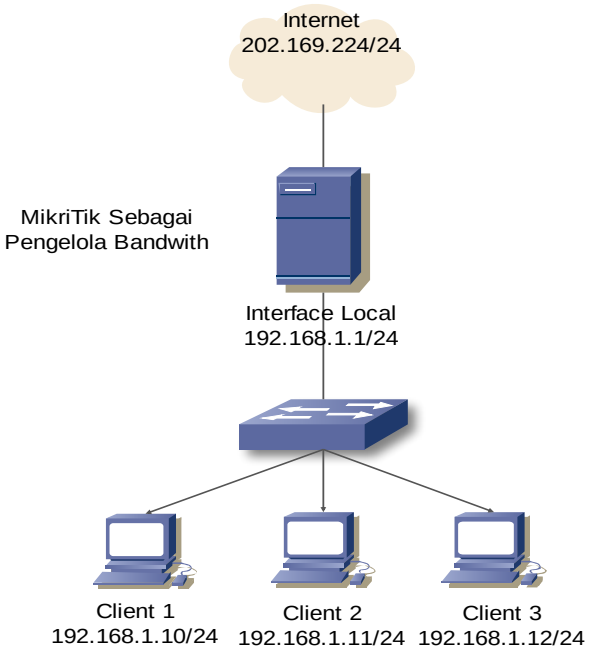
Simple queue adalah cara sederhana melakukan limit date rate untuk IP address atau subnet. Adapun fitur yang dimiliki antara lain :

Peer to peer traffic queueing.

1. Mengizinkan pembuatan aturan queue dengan pemilihan internal waktu.
2. Penggunaan prioritas.
3. Menggunakan multiple paket dengan menggunakan /ip firewall mangle.
4. *Limit traffic* dari dua arah (satu limit untuk total upload + download).

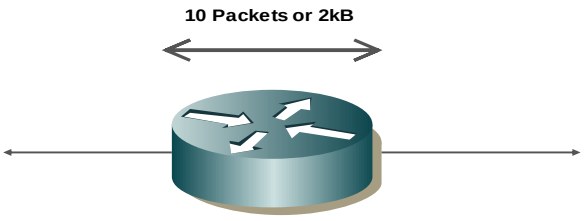
Pada saat menerapkan queue pada jaringan dikenal dua rate atau alokasi bandwith yang akan didapat oleh setiap user, yaitu :

1. *Commited Information Rate* (CIR), merupakan alokasi bandwith terendah yang bisa didapatkan oleh sebuah komputer user jika *traffic* jaringan sangat sibuk. Seburuk apapun keadaan dari jaringan tersebut, komputer user tidak akan mendapatkan alokasi bandwith di bawah dari CIR.
2. *Maximum Information Rate* (MIR), merupakan alokasi bandwith maksimum yang bisa didapatkan komputer user. MIR biasanya akan didapatkan seorang user jika ada alokasi bandwith yang tidak digunakan lagi oleh user lain.



Gambar 2.4 Contoh Simple Queue

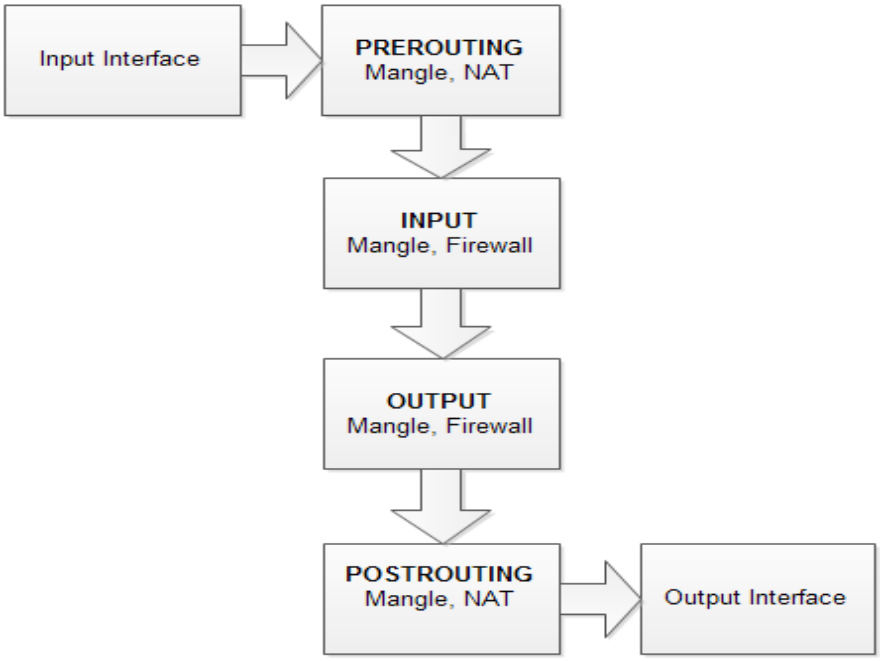
2.4.2 Protokol Layer 7



Gambar 2.5 Layer 7

Protokol Layer 7 bekerja dengan cara mencocokan (*matcher*) 10 paket koneksi pertama atau 2 kilobyte pertama dan mencari pola/*pattern* data yang sesuai dengan yang tersedia. Jika pola ini tidak ditemukan dalam data yang tersedia, *matcher* tidak memeriksa lebih lanjut dan akan dianggap *unknown connections*. *Protokol Layer 7* juga dapat menstabilkan jumlah data yang dikirimkan ke layer 7 sehingga dapat menurunkan penggunaan memori routerboard maupun *Personal Computer* router. *Protokol layer 7* dimanfaatkan secara optimal untuk manajemen bandwith karena filteringnya. *Protokol layer 7* mempunyai metode yang

dipergunakan untuk mendeteksi atau melihat *pattern* didalam suatu koneksi. Pada *protokol layer 7 server* dapat menggunakan *filtering* secara langsung untuk jaringan lokal yang ada. Fitur *protokol layer 7* terdapat pada seri routerboard 4 keatas. *Protokol layer 7* sangat bermanfaat untuk pengaturan manajemen bandwith yang dimana server dapat menentukan konten apa saja yang diberikan speed-limit bandwith oleh router MikroTik. Fitur tambahan *protokol layer 7* ini mempunyai kelebihan untuk *filtering* terhadap *file extension* seperti mp3, rar, zip, mpg, mp4 dan file extension lainnya atau memeriksa aliran paket dari sebuah *interface* (proxy hit).



Gambar 2.6 Traffic Flow

Layer 7 matcher harus melihat kedua arah lalu lintas (masuk dan keluar). Untuk memenuhi persyaratan ini *rule layer 7* harus diatur dalam *chain forward*. Jika *rule* pada chain input/prerouting maka aturan yang sama harus diatur juga dalam *chain output/postrouting*, jika tidak maka data mungkin dianggap tidak lengkap sehingga pola/*pattern* dianggap tidak benar atau tidak cocok. Penggunaan fitur *protokol layer 7* diharapkan dapat membantu sebuah *administrator* untuk dapat mengontrol secara penuh terhadap jaringan lokal yang ditanganinya. Hal ini

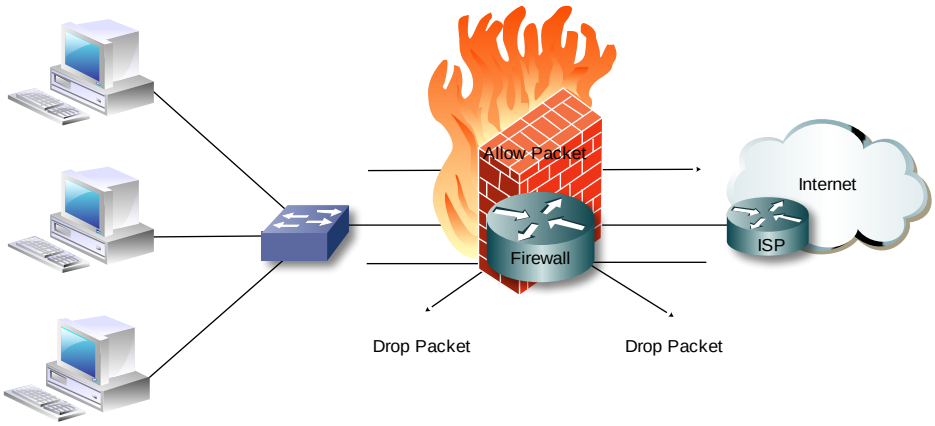
memungkinkan administrator dapat mengontrol secara penuh tiap-tiap *user* yang ada baik untuk bandwith yang telah diberikan maupun *filtering* konten yang tidak diinginkan yang masuk terhadap jaringan lokal.

2.5. *Regular Expression*

Regular Expression merupakan suatu notasi fleksibel dan ringkas untuk mencari dan menggantikan pola teks. Fungsi utama regular expression itu sendiri adalah mencari dan mengganti pola text. Notasi *Regular Expression* terdiri dari dua jenis karakter dasar, yaitu karakter teks literal (normal) dan metakarakter.Karakter normal menyatakan bahwa teks harus eksis di string target, sedangkan metakarakter menyatakan teks dapat bermacam-macam di string target. *Regular Expression* memungkinkan kita menguraikan sejumlah teks guna menemukan pola karakter spesifik. Selain itu, Anda juga bisa mengganti, memodifikasi, atau menghapus suatu substring dengan cepat dan akurat, sesuai kriteria pola yang kita inginkan. Hampir semua bahasa pemrograman mengimplementasikan *regular expression*.

2.6 Firewall

Firewall adalah program atau kumpulan program yang berada pada server gateway jaringan yang melindungi jaringan lokal dari pengguna jaringan lain (Internet). Firewall berkerjasama dengan program router untuk memeriksa semua data yang lewat untuk menentukan apakah data tersebut akan dilewatkan ketujuannya atau ditolak. Biasanya firewall diinstall dalam *server* yang menjadi gateway atau penghubung antara dua jaringan.Berikut sebuah ilustrasi firewall yang membatasi akses *user* ke *server* jaringan.



Gambar 2.7 Firewall

Firewall digunakan untuk membatasi akses yang berasal dari jaringan external (misalnya internet) dari berbagai ancaman yang sering kali berasal dari internet. Firewall berfungsi menjaga keamanan jaringan dari ancaman pihak lain yang tidak berwenang. Mengubah, merusak, atau menyebarkan data-data penting perusahaan merupakan contoh ancaman yang harus dicegah. Firewall beroprasi menggunakan aturan tertentu. Aturan inilah yang menentukan kondisi ekspresi yang memberitahu router tentang apa yang harus dilakukan router terhadap paket IP yang melewatinya. Setiap aturan disusun atas kondisi dan aksi yang akan dilakukan. Ketika ada paket IP lewat, firewall akan mencocokkannya dengan kondisi yang telah dibuat kemudian menentukan aksi apa yang akan dilakukan router sesuai dengan kondisi tersebut.

Mikrotik juga dipadukan dengan kemampuan firewall untuk mencegah hal-hal yang menggangu dari pihak lain, mengingat begitu banyaknya aplikasi yang dijalankan oleh pengguna jaringan. Ada aplikasi yang berjalan normal, tetapi ada juga aplikasi yang bersifat menggangu kinerja jaringan. Sebagai contoh, paket broadcast yang dilakukan oleh virus dan paket yang berlebihan atau sering disebut sebagai *flooding*.

Paket dengan ukuran kecil memang tidak menggangu koneksi jaringan. Namun, jika paket yang kecil tersebut dalam jumlah banyak, hal ini bisa menurunkan kinerja jaringan (*down*). Maka disinilah pentingnya menggunakan firewall untuk

menghindari insiden yang bersifat negatif dan membuat jaringan yang ada tetap stabil. Pada sistem operasi MikroTik, firewall sudah termasuk paket MikroTik RouterOS™ yang di dalam direktori firewall sendiri terdapat 6 direktori :

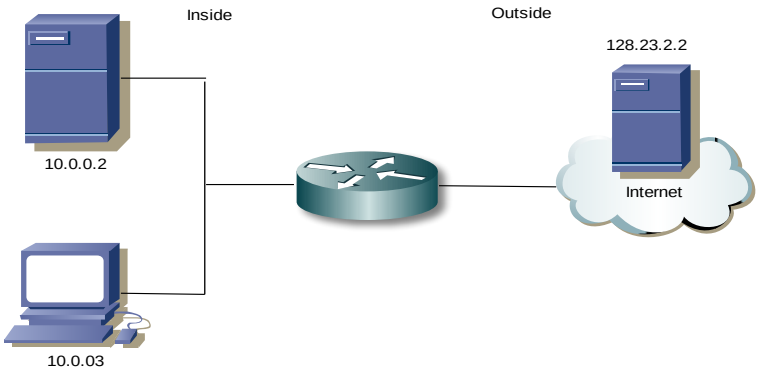
1. ***Mangle***, untuk menandai paket dengan suatu tanda khusus sebagai identitas paket tersebut.
2. ***NAT***, untuk memetakan IP address ke IP address lain.
3. ***Connection***, untuk mengetahui informasi dari suatu koneksi yang aktif seperti IP address asal dan tujuan koneksi beserta port yang digunakan, jenis protokol yang dipakai.
4. ***Address-list***, untuk mendefenisikan IP address ke dalam group tertentu.
5. ***Service-port***, untuk mengaktifkan dan mengubah nomor port aplikasi.
6. ***Filter***, untuk memfilter paket yang masuk atau melewati router. Router akan meneruskan jika paket diizinkan lewat dan sebaliknya.
7. ***Export***, untuk menyimpan/backup semua konfigurasi di dalam direktori firewall.

2.6.1 Firewall NAT (SRC-NAT)

Network Address Translation (NAT) adalah suatu fungsi firewall yang sebenarnya bertugas melakukan perubahan IP address pengirim dari sebuah paket data. NAT ini umumnya dijalankan di router-router yang menjadi batas antara jaringan lokal dan jaringan internet. Secara teknis NAT ini akan mengubah paket data yang berasal dari komputer user seolah-olah berasal dari router. Sebuah komputer yang akan melakukan koneksi ke internet harus menggunakan IP Public, namun dengan semakin banyaknya penggunaan internet, jumlah IP Public yang tersedia semakin lama semakin berkurang, maka munculah metode NAT (*Network Address Transaction*), Metode ini akan mentranslasikan IP Privte jaringan lokal menjadi IP Public, sehingga dapat terkoneksi ke jaringan internet.

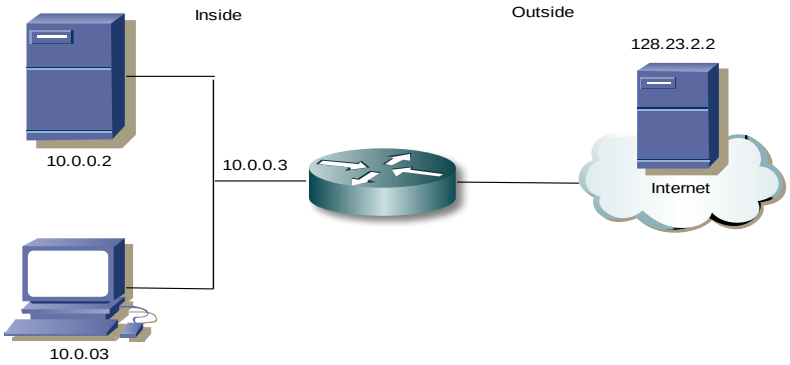
Berikut merupakan gambaran terjadinya proses NAT :

Jaringan lokal menggunakan akses kesebuah server internet dengan alamat 128.23.2.2



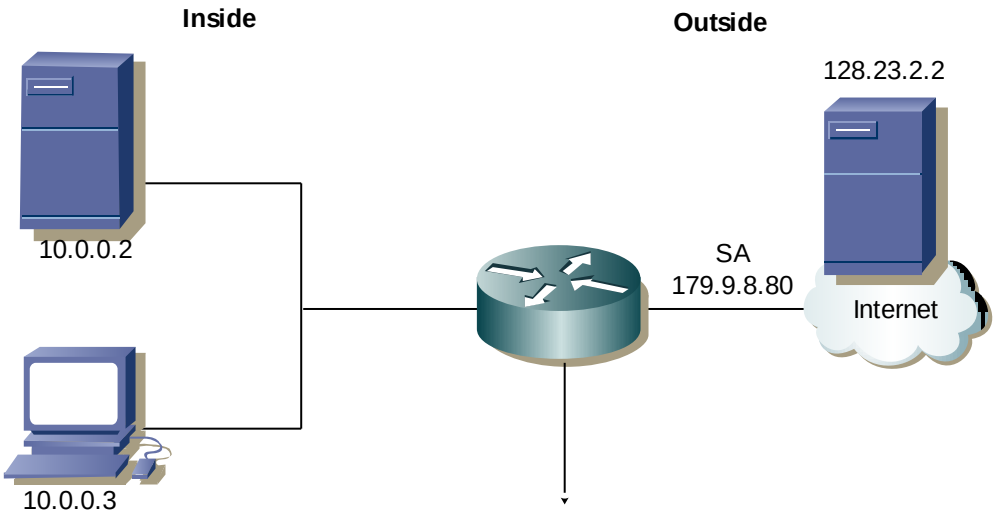
Gambar 2.8 Proses Firewall NAT

Paket data akan masuk kedalam router dan dilakukan proses NAT



Gambar 2.9 Proses Firewall NAT 2

Setelah melalui proses NAT maka IP address jaringan lokal yang awalnya 10.0.0.2 akan ditranslasikan menjadi IP Public 172.9.8.80.



NAT Table		
Inside Local IP Adress	Inside Global IP Address	Outside Global IP Address
10.0.0.3	179.9.8.80	128.23.2.2

Gambar 2.10 Proses Firewall NAT 3

BAB III
METODOLOGI PENELITIAN

3.1 Analisis Masalah

Data dari sebuah studi bertema *global bandwith study* yang dilakukan oleh CIP Technologies, menunjukan bahwa pengguna internet didunia terancam kelaparan bandwith. Hal ini terjadi ketika infrastruktur yang ada tidak mampu melayani permintaan bandwith untuk melayani semua keinginan konsumen dalam mengakses internet (CIP Technologies, 2008).

Untuk mengatasi hal tersebut berbagai cara telah diupayakan agar bandwith yang ada dapat terpenuhi oleh setiap pengguna, salah satu upaya tersebut adalah mengatur bandwith dengan manajemen bandwith yang tepat. Manajemen bandwith menggunakan *simple queue* dan *queue tree* dengan *protokol layer 7* yang telah ada dalam mikrotik adalah salah satu cara untuk pegaturan bandwith dalam sebuah infrastruktur jaringan. Manajemen bandwith yang tepat pada sebuah jaringan akan memaksimalkan jumlah bandwith yang ada.Pengguna manajemen bandwith *simple queue* dan *queue tree* dengan *protokol layer 7* pada iTCentrum membantu dalam pengaturan bandwith pada setiap komputer siswa pelatihan dan memberikan *client* bandwith yang merata dan sesuai kebutuhan internet mereka.

Perancangan sistem menggunakan queue tree dengan *protocol layer 7* juga bertujuan untuk menjaga keamanan dan kenyamanan dalam jaringan. Perancangan sistem *firewall protocol layer 7* agar dapat menjadikan jaringan internet yang ada menjadi sebuah internet yang sehat. Pengimplimentasian internet sehat menggunakan MikroTik diharapkan dapat membantu mengatasi masalah selain penggunaan bandwith yang berlebihan juga untuk mengatur *file-file* yang tidak diinginkan masuk kedalam jaringan tersebut.

Untuk mengetahui apakah metode manajemen bandwith menggunakan *simple queu*e dan *queue tree* dengan *protokol layer 7* sudah tepat digunakan dalam jaringan iTCetrum apalagi untuk jaringan Internet kampus, maka dilakukan analisi terhadap

fitur *quality of service* yang mana yang lebih baik untuk pengaturan manajemen bandwith disuatu tempat.

3.2 Analisis Kebutuhan Sistem

3.2.1 Kebutuhan perangkat keras

Gambaran umum perangkat keras yang dibutuhkan untuk menunjang implementasi dalam menjalankan sistem diantaranya :

1. Komputer sebagai server

   Komputer ini merupakan komputer yang difungsikan sebagai *gateway* dan terdapat *simple queue* serta *protokol layer 7* yang menuju ke internet dari jaringan lokal dan sebagai bandwith manajemen untuk jaringan lokal. Berikut spesifikasi komputer yang dibutuhkan :

   a. Prosesor AMD Turion™ II Dual-Core Mobile M520 (2 CPUs), ~2.3GHz
   b. Memori 4GB
   c. Harddisk dengan kapasitas 320GB
   d. Network Interface Card 2 buah

2. Komputer sebagai *client*

   Komputer yang digunakan untuk pengujian

   a. Prosesor AMD Turion™ II Dual-Core Mobile M520 (2 CPUs), ~2.3GHz
   b. Memori 4GB
   c. Harddisk dengan kapasitas 320GB
   d. Network Interface Card 2 buah

3. MikroTik Router RB493 Bertugas sebagai *gateway* langsung dari *server* dan *client* dan sebagai bandwith management untuk jaringan lokal. Routerboard ini merupakan router yang didalamnya sudah terdapat OS yang siap dipakai. Routerboard dignakan adalah seri RB493 yang memiliki 5 port Ethernet. Berikut detail spesifikasi RB493:

   a. CPU Speed 600MHz

b. Memori 60.7 MiB
c. Storage 61.4 MB
d. Bios Version 5.7

3.2.2 Kebutuhan Fungsionalitas

Kebutuhan fungsi menunjukan peran fungsi dalam aplikasi sebagai bentuk dari rancangan aplikasi yang dibangun. Fungsi-fungsi yang dibutuhkan tersebut sebagai berikut :

1. Fungsi *Speed-limit*

   Fungsi ini digunakan untuk melakukan proses *speed-limit* unduh maupun unggah yang akan ditampilkan dalam sistem dan sebagai tolak ukur perhitungan, sehingga saat menjalankan manajemen bandwith pengguna dapat melakukan proses pembagian bandwith.

2. Fungsi *Drop* (*Blocking*)

   Fungsi ini digunakan untuk melakukan proses *drop* terhadap paket-paket yang akan masuk ke dalam router, sebagai batasan terhadap akses yang diberikan sistem untuk proses penggunaan jaringan internet di iTCentrum.

3.2.3 Kebutuhan Antar Muka

3.2.3.1 Kebutuhan Antar Pengguna

1. Windows

   Winbox merupakan sebuah aplikasi yang memungkinkan kita melakukan konfigurasi MikroTik RouterOS™ menggunakan *graphic interface* yang cepat dan sederhana.WinBox merupakan aplikasi biner Win32 (Windows).Semua fungsi interface WinBox dibuat sedekat mungkin fungsi *console*, itu sebabnya tak ada bagian WinBox dimanual. Terdapat beberapa konfigurasi tingkat *advanced* dan konfigurasi sistem paling tidak dapat dilakukan dari WinBox, seperti perubahan MAC Address pada sebuah atarmuka atau mem-factory default Mikrotik. Oleh karena itu,

dalam software antarmuka pengguna mengacu pada antarmuka windows dan konfigurasi pada umumnya.

2. Linux dan Mac OS

   Untuk sistem operasi linux dan Mac OS aplikasi winbox tetap dapat dijalankan dengan mnggunakan program bantu *Wine*. *Wine* adalah sebuah software yang memungkinkan menjalankan program Windows pada sistem operasi Unix-like di arsitektur x86 apapun, terutama Linux. Wine juga menyediakan sebuah software library yang dikenal sebagai *Winelib* yang dapat membantu developer meng-*compile* aplikasi Windows dan memportkannya ke sistem Unix-like.

3.2.3.2 Kebutuhan Antar Muka Eksternal

Merupakan kebutuhan antar muka perangkat yang digunakan oleh admin dalam proses pembuatan konfigurasi yang akan dilakukan.

1. Komputer PC

   Merupakan komponen utama yang digunakan dalam pengoprasian perangkat lunak yang terdiri dari CPU, monitor, mouse, keyboard.

2. Router

   Merupakan komponen utama pendukung yang digunakan dalam kegiatan pengoprasian penggunaan dan konfigurasi jaringan yang akan dijalankan.

3.2.3.3 Kebutuhan Antar Muka Perangkat Lunak

1. Windows 7

   Merupakan sistem operasi yang sangat umum digunakan, pengguna windows dikarenakan spesifikasi yang lebih efisien dan terdapat banyak aplikasi pendukung untuk melakukan penelitian ini.

2. MikroTik RouterOS™

   MikroTik RouterOS™ merupakan sistem operasi dengan perangkat lunak yang dapat digunakan pada komputer PC menjadi sebuah

komputer router network yang handal.Mencakup berbagai fitur yang dibuat untuk *IP network* dan jaringan *wireless*. Mikrotik didesain untuk memberikan kemudahan bagi penggunanya.Administrasinya bisa dilakukan melalui *Windows Application* (WinBox). PC yang akan dijadikan router mikrotik pun tidak memerlukan *resource* yang cukup besar untuk penggunaan standar, misalnya hanya sebagai gateway. Mikrotik RouterOS™ yang berbentuk software yang dapat diunduh di www.mikrotik.com merupakan jenis MikroTik OS yang dapat diinstal pada komputer.Sedangkan BUILT-IN Hardware Mikrotik merupakan jenis MikroTik dalam bentuk perangkat keras yang khusus dikemas dalam board router yang didalamnya sudah terinstal MikroTik RouterOS™.

3. Winbox

   WinBox merupakan sebuah aplikasi yang memungkinkan kita melakukan konfigurasi MikroTik RouterOS™ menggunakan *graphic interface* yang cepat dan sederhana. WinBox merupakan aplikasi biner Win32 (Windows), WinBox berjalan pada sistem operasi Windows, akan tetapi dapat dijalankan pada sistem operasi Linux dan Mac OS dengan menggunakan program bantu Wine. Semua fungsi interface WinBox dibuat sedekat mungkin fungsi *console*, itu sebabnya tak ada bagian WinBox dimanual. Terdapat beberapa konfigurasi tingkat *advanced* dan konfigurasi sistem paling tidak dapat dilakukan dari WinBox, seperti perubahan MAC Address pada sebuah atarmuka atau mem-factory default Mikrotik.

4. Wireshark

   Wireshark merupakan salah satu tools atau aplikasi "*Network Analyzer*" atau Penganalisa Jaringan. Penganalisaan Kinerja Jaringan itu dapat melingkupi berbagai hal, mulai dari proses menangkap paket-paket data atau informasi yang berlalu-lalang dalam jaringan, sampai pada digunakan pula untuk *sniffing* (memperoleh informasi penting

seperti password email, dll). Wireshark sendiri merupakan *freetools* untuk *Network Analyzer* yang ada saat ini. Dan tampilan dari wireshark ini sendiri terbilang sangat bersahabat dengan user karena menggunakan tampilan grafis atau GUI (*Graphical User Interface*).

5. Filezilla

   Filezilla atau juga dikenal sebutan FileZilla Client, adalah salah satu software FTP (*File Transfer Protocol*) berfungsi untuk mengupload atau mendownload file sebuah website, Filezilla juga merupakan salah satu program Software Open Resource yang bisa kita dapatkan di file-zilla-project.org

6. Command Prompt

   Merupakan user interface berbasis text yang dipergunakan untuk mengeksekusi baris perintah yang dimasukkan pada sistem operasi windows.

7. Graph tool

   Tool ini berfungsi untuk memantau trafik jaringan pada interface, namun graping juga bisa memantau trafik resource yang terpakai oleh router seperti cpu, memory dan hardisk.

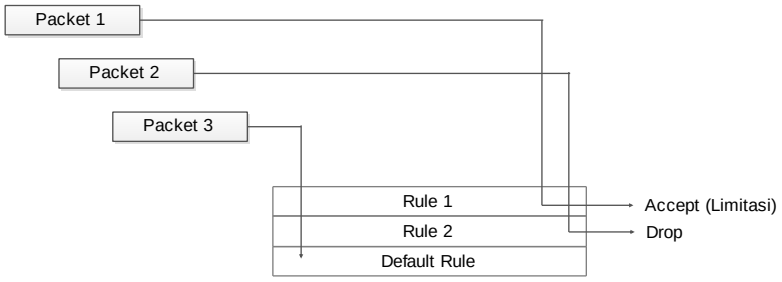
3.3 Gambaran Umum Sistem

Manajemen bandwidth dalam suatu jaringan merupakan proses dimana seorang *administrator* melakukan semua kegiatan untuk mengontrol, membatasi, mengawasi sebuah jaringan yang dipegang agar dapat digunakan sesuai aturan yang ada di iTCentrum dan aturan untuk pengguna jaringan tersebut. Dalam sistem ini administrator mencoba untuk melakukan proses pembatasan maupun penolakan paket yang akan diminta oleh pengguna internet tersebut. *Sistem queue* dan *queue tree* menggunakan *firewall protokol layer 7* dijalankan melalui winbox yang menjadi interface untuk melakukan semua proses yang akan dilaksanakan.

Konsep *filtering* digunakan untuk menentukan paket apa saja yang bisa masuk maupun keluar dari jaringan. Untuk menentukan paket mana yang akan diterima

(*accept*) atau dibuang (*drop*), firewall akan memeriksa *header* dari sebuah IP *packet*. Sistem Firewall ini melakukan pemeriksaan pada isi dari data yang dikirimkan.

Paket yang masuk ke dalam firewall akan dicocokan dengan *rule-rule* yang sudah dibuat. Bila ada paket yang cocok dengan *rule* yang sudah dibuat, maka barulah firewall akan menentukan tindakan (*action*) apa yang akan diambil, apakah paket tersebut akan diterima (*accept*) kemudian dilimitasi atau akan dibuang (*drop*). Gambaran antara paket dan rule firewall dapat dilihat pada gambar berikut.



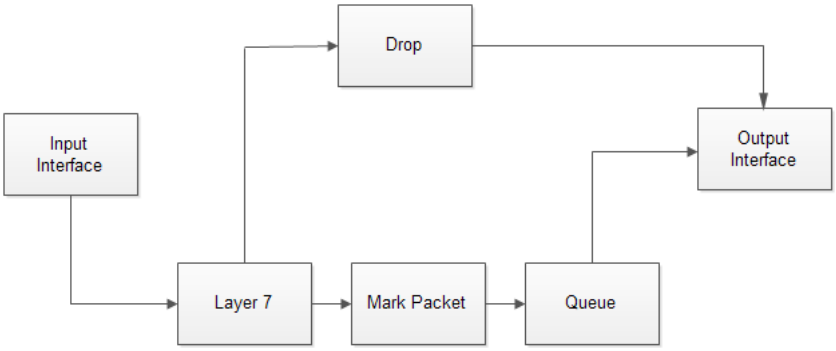
Gambar 3.1 Proses Rule Paket

Pada gambar 3.1 diatas, ada 3 buah paket yang akan diproses oleh sistem pada router mikrotik. Router mikrotik tersebut telah memiliki 2 buah *rule* dan sebuah rule default. Dibawah ini merupakan proses perlakuan dari masing-masing paket tersebut.

1. Paket 1, pada saat paket ini diterima, firewall menggunakan *Rule* 1 untuk memeriksanya. Ternyata header pada paket 1 cocok dengan rule dengan parameter rule yang ada pada *rule* 1, maka paket akan secara langsung diterima oleh *rule* 1 dan disinilah queue dan firewall akan mengambil *action accept* dan langsung dilimitasi oleh sistem.
2. Paket 2, dapat dilihat bahwa header dari paket tidak cocok dengan rule 1 maka paket akan diteruskan ke *rule* 2 untuk diperiksa lebih lanjut apakah header paket 2 masuk kedalam *rule* 2 yang memiliki *action drop* sesuai dengan sistem queue dan firewall yang telah dikonfigurasi. Hasilnya ternyata *header* paket 2 cocok dengan *rule* yang telah dibuat yaitu paket harus dibuang (*drop*) oleh sistem.

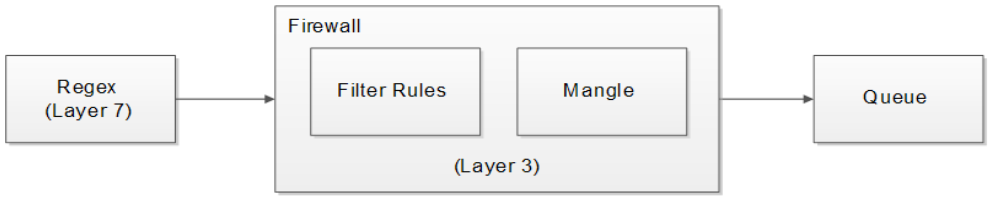
3. Paket 3, paket ini merupakan paket dengan header yang tidak memiliki kecocokan dengan *rule* yang ada. Dengan kata lain, tidak ada *rule* yang berhasil memeriksa paket 3. Jika ini terjadi, maka paket akan diproses di *default rule*. Pada router mikrotik, *default rule* yang digunakan adalah *accept all*, artinya bila ada paket yang tidak cocok dengan *rule* yang ada sebelumnya, maka paket tersebut akan tetap berjalan tanpa melalui rule yang ada.

Secara keseluruhan sistem yang dirancang untuk jaringan iTCentrum merupakan perpaduan antara queue dan firewall yang ada pada router mikrotik. Gambaran secara menyeluruh dari sistem yang dibuat dapat dilihat pada gambar berikut.



Gambar 3.2 Proses Sistem

Pada gambar 3.2 diatas merupakan proses sistem queue dan firewall yang digunakan untuk melakukan *speed-limit* dan *drop* dari paket yang akan masuk ke dalam jaringan tersebut.

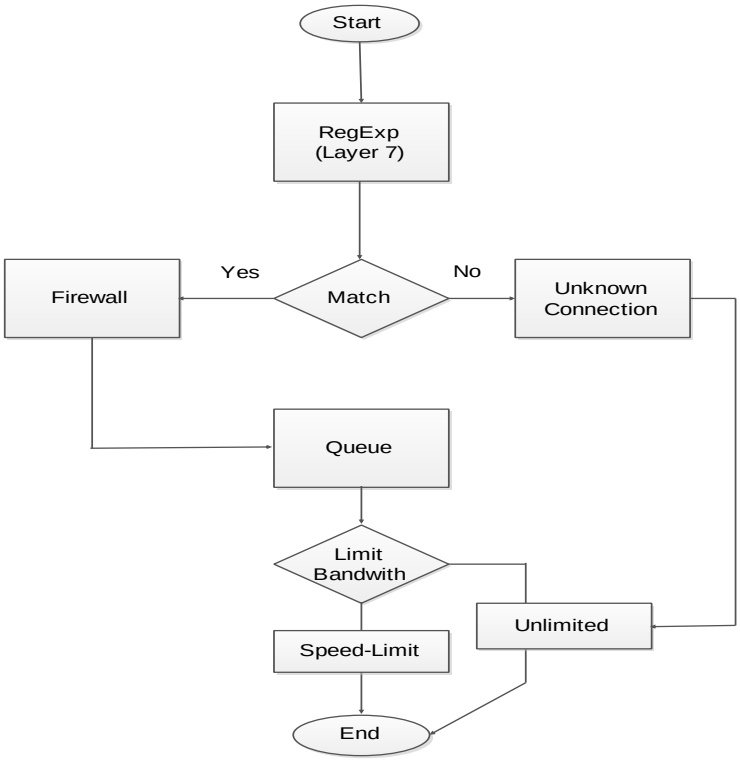


Gambar 3.3 Diagram Komponen

Pada gambar 3.3 merupakan diagram komponen skenario firewall pada konfigurasi queue tree. Proses pertama yaitu ReGex (*Layer 7*), proses kedua yaitu firewall (filter rules dan firewall mangle), dan yang terakhir adalah proses *queue*.

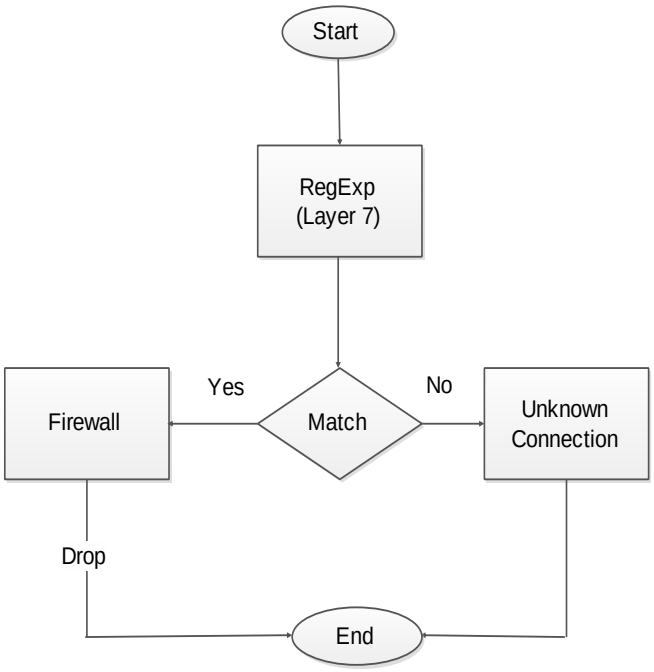
Gambar 3.4 Flowchart Default Rule

Gambar 3.4 merupakan gambar *default rule* dari proses *replay* paket yang ada pada konfigurasi mikrotik. Start (paket) akan langsung kedalam Konfigurasi belum menggunakan firewall sehingga pada proses ini paket akan mendapatkan bandwith secara *unlimited* pada saat terjadi proses pengunduhan.



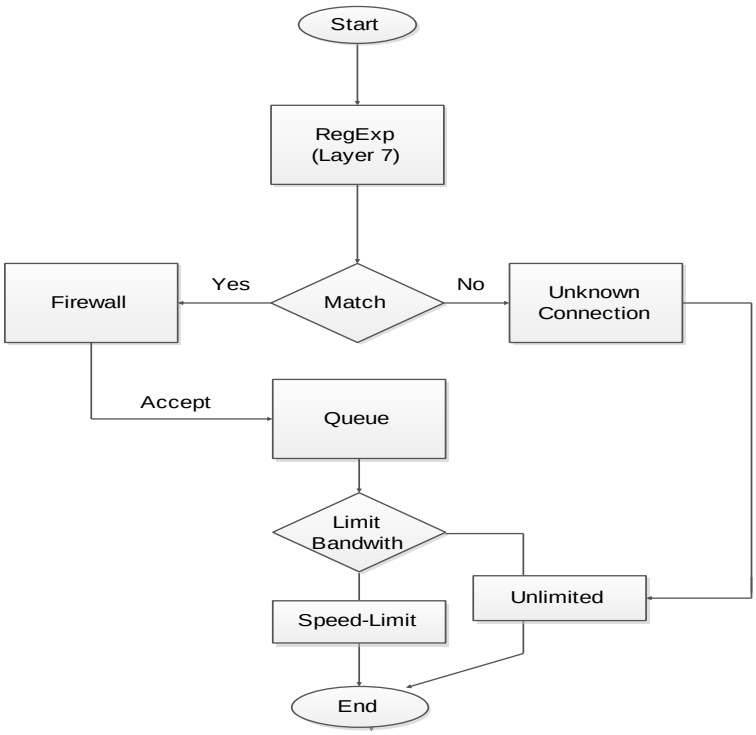
Gambar 3.5 Flowchart Sistem

Pada gambar 3.5 ditambahkan merupakan penelitian yang nantinya akan saya lakukan konfigurasi ditambahkan *regular ekspression* yang memeriksan header paket, firewall yang akan melakukan *action* dan menandai paket serta queue untuk bandwith pengunduhan paket dengan limit bandwith *speed-limit* atau *unlimited.*



Gambar 3.6 Flowchart *Drop*

Pada gambar 3.6 proses masuknya paket ditandai dengan start yang kemudian akan diteruskan pada pemeriksaan paket di RegExp dimana paket akan dicocokan, pada proses pencocokan apabila paket sesuai (*yes*) akan dilanjutkan ke firewall yang kemudian ketika paket dibuang (*drop*) akan langsung ke selesai (*end*). Pada proses pencocokan (*no*) maka dianggap *unknown connection* paket juga akan selesai (*end*).



Gambar 3.7 Flowchart *Accept*

Pada gambar 3.7 merupakan flowchart *accept* proses masuknya paket (*start*) kemudian akan diteruskan ke proses pemeriksaan paket (RegExp), didalam proses RegExp paket akan ditentukan apakah sesuai atau tidak, paket susuai (*yes*) maka akan diteruskan lagi ke dalam aturan firewall dengan *action* (*accept*) yang akan diteruskan ke proses selanjutnya yaitu *queue* dengan pilihan limit bandwith dengan proses *speed-limit* kemudian selesai (*end*) atau unlimited untuk *unknown connection* yang merupakan pilihan sebelumnya dari regular ekspresion matching (*no*) dan dianggap selesai (*end*).

3.3.1 *RegularExpression*

Seperti yang sudah dibahas sebelumnya penggunaan Regexp Protocol Layer 7 mencoba untuk melakukan *bloking* koneksi berdasarkan *pattern*, karena jika menggunakan *content* kadang akan kesulitan untuk melakukan *blocking content* tersebut. Regular ekspression sendiri digunakan oleh banyak teks editor, utilities, dan

bahasa pemrograman untuk pencarian dan manipulasi teks berdasarkan pola.Penggunaan regular expression sendiri memang lebih banyak dikembangkan untuk program-program yang mendukung adanya pola teks seperti PHP, perl, ruby dan lain-lain.

Tidak hanya untuk programming ternyata MikroTik juga mengembangkan sistem pola teks ini kedalam router-router MikroTik. Pengembangan penggunaan firewall protocol layer 7 untuk mendukung mengingkatkan implementasi keamanan jaringan yang ada pada MikroTik. Untuk itulah sistem jaringan yang dibuat di iTCentrum menggunakan salah satu fitur yang ada pada firewall yaitu protocol layer 7 yang sekaligus digabungkan dengan manajemen bandwith queue.

Regular Expression yang peneliti gunakan dapat dilihat pada tabel 3.1 dibawah.

Tabel 3.1 *RegularExpression* Firewall Protocol Layer 7 MikroTik

RegExp	Keterangan
^ (Caret / Tanda Sisispan)	Awal Masukan
$	Akhir Input
.	Karakter
* (Star)	0 atau lebih ekspresi sebelumnya
+	1 atau lebih ekspresi sebelumnya
\	Mendahului salah satu diatas, membuat literal bukan karakter khusus.
( )	Gruping, digunakan untuk mengelompokan karakter-karakter menjadi *single* unit.

Banyak cara penulisan yang digunakan dalam *regular expression.* Salah satu cara penulisan *regular expression* yang digunakan dalam sistem yang dirancang peneliti adalah \\.(mp3) karena lebih sederhana dan mudah dipahami sebagai tingkatan dasar. Tanda \ digunakan untuk mengembalikan fungsi metacharacter menjadi karakter biasa atau sebaliknya. Tanda ^ merupakan awal masukan atau penanda awal sebuah string. Tanda $ merupakan akhir input atau penanda akhir dari

sebuah string. Tanda . merupakan pendefenisian sebuah karakter. Tanda * 0 atau lebih ekspresi sebelumnya sebagai contoh pattern “te*s” cocok dengan string “ts”, “tes”, “tees” dan seterusnya. Tanda + 1 atau lebih ekspresi sebelumnya sebagai contoh “ayu+k” cocok dengan “ayuk”, “ayuuk”, “ayuuuk” dan seterusnya. Tanda ( ) merupakan gruping atau pengelompokan karakter.Itulah beberapa pendefinisian yang digunakan untuk sebuah regular expression.

Untuk file-file ekstensi sendiri yang penulis deskripskan pada sistem yaitu \\.(file ekstension). Dalam sistem yang dibuat penulis mengambil beberapa sampel file ekstensi yang dijadikan implementasi dan pengujian firewall protocol layer 7. File ekstensinya antara lain mp3, exe, zip, dan flv. Jadi untuk mendeskripsikan file ekstensi tersebut kedalam regexp menjadi “\\.(mp3)” , \\.(exe), \\.(zip), dan \\.(flv). File ekstensi diatas yang nantinya akan diterima (*accept)* dan dibuang (*drop)* pada pengujian. Regular expression tentu akan dapat dikenali apabila penulisan regexp sesuai dengan aturan penulisan yang sesuai, apabila penulisan regexp salah maka sistem tentu tidak akan bisa membaca atau mengenali pola tersebut.

3.4 Metode Pengembangan Sistem

Metode pengembangan sistem dilakukan secara berurutan untuk sistem yang dibangun.Metode-metode pengembangan sistem tentu sangat membantu penulis dalam menganalisa dan sampai pada tahap untuk pemeliaraan sistem tersebut. Adapun tahapan-tahapan yang nantinya akan dilakukan untuk membangun sistem ini antara lain :

1. Analisis Kebutuhan

   Kebutuhan yang diperlukan untuk sistem ini didapat dari data-data sebelumnya yang ada pada iTCentrum. Analisis semua yang berubungan dengan sistem juga perangkat-perangkat lunak yang akan digunakan guna membantu dalam menyusun sistem manajemen bandwith menggunakan simple queue dan queue tree dengan firewall protokol layer 7 yang diinginkan.

2. Desain Sistem

   Untuk sistem yang dirancang disesuaikan dengan topologi yang juga nantinya akan dibuat di iTCentrum. Sistem sendiri dibangun sesuai dengan proses dan rule-rule dari masing-masing sistem manajemen bandwith.

3. Pembangunan Sistem

   Pembangunan sistem dilakukan dengan mengkonfigurasi dan menginstall semua perangkat-perangkat lunak pendukung yang sesuai dengan kebutuhan sistem manejemen bandwith simple queue dan queue tree protokol layer 7 pada iTCentrum

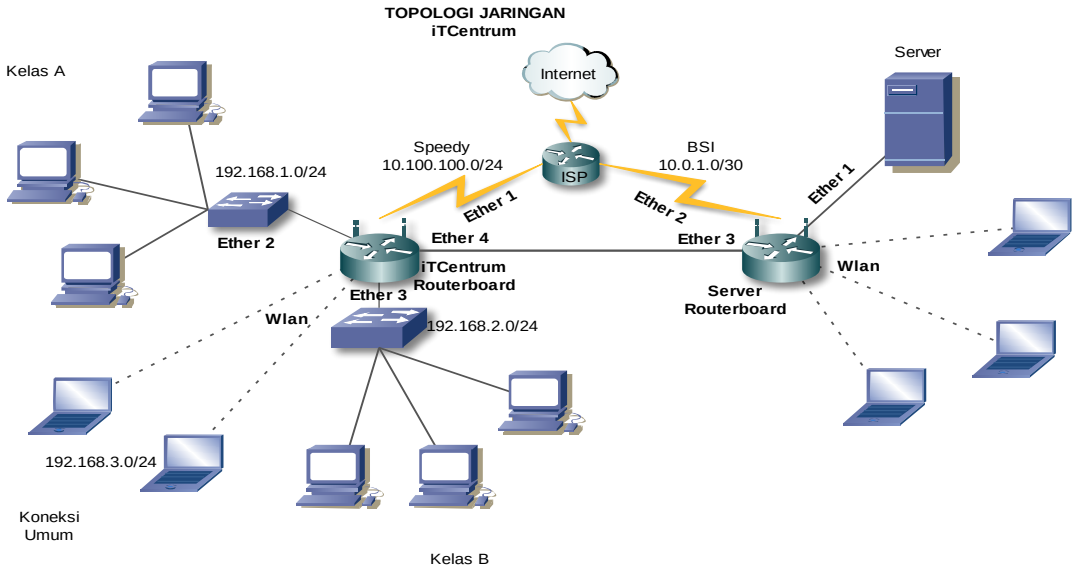
4. Pengujian Program

   Ini merupakan tahapan akhir dari sistem yang telah dibuat.Pada tahapan ini sistem manajemen bandwith simple queue dan queue tree di uji kinerja dan fungsionalitasnya yang juga menyangkut kemampuan serta keefektifann dari sistem sehingga dapat diperoleh kekurangan dan kelemahan dari sistem manajemen bandwith ini. Hal ini juga dilakukan agar apabila terdapat masih banyak kekurangan dalam sistem akan dilakukan perbaikan agar sistem menjadi lebih baik dan sempurna.

3.4.1 Topologi

IT Centrum adalah salah satu pusat pelatihan dan sertifikasi berskala Nasional maupun Internasional, berbagai macam pelatihan yang disediakan oleh IT Centrum adalah Adobe, CISCO, EC Council, Java, dan Oracle. IT Centrum terletak di Kampus FTI terpadu Universitas Islam Indonesia.Sebagai tempat pelatihan yang berada di lingkungan kampus yang berkembang sangat cepat, hal ini membuat IT Centrum di tuntut mampu menyesuaikan dengan kebutuhan para *trainer*, baik itu dalam hal pemberian materi, maupun kemudahan akses internet, serta kenyamanan mengakses internet. iTCentrum menggunakan jaringan speedy untuk jaringan iTCentrum yang ada dibawahnya dan Server iTCentrum yang digunakan untuk semua

keperluan materi pelatihan, database mahasiswa yang ikut pelatihan serta trainer. iTCentrum memiliki bandwith sebesar 2048 kbps (*Kilobit per second*) untuk menunjang semua aktivitas koneksi upload dan download. Berikut merupakan gambar topologi yang ada di iTCentrum :



Gambar 3.8 Topologi Jaringn iTCentrum

Pada gambar 3.8 topologi iTCentrum terdepat beberapa pembagian ip address untuk masing-masing client ataupun server yang ada pada pusat pelatihan sertifikasi mahasiswa. Kelas A mendapatkan ip address 192.168.1.0/24 (ether2), kelas B mendapatkan ip address 192.168.2.0/24 (ether3), koneksi umum yang merupakan koneksi hotspot yang diberikan untuk mahasiswa *client* di luar komputer yang telah ada mendapatkan ip address 192.168.3.0/24 (Wlan1) sedangkan untuk jaringan internet sendiri iTCentrum menggunakan koneksi speedy dengan ip address 10.100.100.0/24 (ether1), serta koneksi server 10.0.1.1/30 (ether4 routerboard iTCentrum).

Pada gambar 3.8 juga menjelaskan bahwa iTCentrum menggunakan satu routerboard lagi untuk tambahan koneksi BSI dan koneksi server iTCentrum itu sendiri. Koneksi BSI dan server iTCentrum dihubungkan kedalam jaringan

iTCentrum agar mahasiswa maupun dosen dapat menggunakan server iTCentrum untuk berbagai aktivitas upload maupun download untuk materi-materi dan file-file yang berhubungan dengam pelatihan itu sendiri. Koneksi BSI mendapatkan ip address 10.0.1.0/30 (ether2), PC server mendapatkan alamat IP 10.0.1.1/30 (ether3 routerboard server).

3.4.2 Rule dan Policy

Sistem dibuat sesuai dengan aturan yang telah berlaku pada MikroTik RouterOS™. Penggunaan queue dan firewall yang dibuat tetap mengikuti aturan-aturan yang telah diterapkan oleh mikrotik. ***Rule*** merupakan kebijakan yang berkaitan dengan bagaimana aturan akan dilaksanakan. *Rule-rule* yang diterapkan :

1. *Rule* untuk *admin*
    a. Administrator tidak memberikan hak akses khusus untuk user tertentu
    b. Administrator menambah dan mengubah rule sesuai dengan kebutuhan yang diperlukan.
2. *Rule* untuk *user*
    a. Sistem menyediakan bandwith yang terbatas
    b. Sistem memberikan pembatasan hak download terhadap file ekstensi yang ingin diunggah
    c. Sistem tidak bisa diubah kecuali mendapatkan perijinan dari administrator
3. *Rule* untuk *sistem*
    a. Sistem dibuat tetap berdasarkan aturan yang berlaku pada MikroTik RouterOS™.

Aturan-aturan yang ada pada sistem diterapkan untuk mengatur semua aktifitas yang ada pada sistem baik dari sistem sendiri maupun terhadap admin maupun user.

Policies bagaimana aturan itu akan dilaksanakan. Kebijakan ini umumnya mengarahkan kepada bagaimana mencapai tujuan dari aturan yang diterapkan. *Rule* administrator diharapkan dapat memberikan arahan untuk *admin* bagaimana

menjalankan sistem yang telah dirancang. *Rule user* memberikan arahan kebijakan yang telah diterapkan agar dapat dipatuhi oleh semua pengguna tanpa terkecuali. *Rule* untuk sistem merupakan standar dari aturan-aturan yang telah ada dan agar sistem dibuat sesuai dengan standarisasi yang diterapkan oleh mikrotik sebelumnya tanpa mengubah bentuk dari standarisasi tersebut.

3.5 Metode Pengujian Sistem

Metode analisis yang dilakukan dalam penelitian ini adalah dengan statistik deskriftif dan induktif terhadap hipotesis yang telah diberikan di awal penelitian. Data yang akan di analisis adalah fungsional dan kinerja baik berupa *throughput*, *delay*, *packet loss* pada pengambilan data menggunakan filezilla, wireshark, maupun protokol UDP.

1. Stastik Deskriptif

   Statistik deskriptif yaitu dengan pengumpulan, pengolahan, dan penyajian data baik dalam bentuk tabel, grafik, maupun diagram.Statistik deskriptif memberikan informasi yang digunakan untuk menjawab rumusan masalah yang ada.

Sebelumnya sudah dijelaskan pada penelitian ini menggunakan metode eksperimen untuk mendapatkan parameter pengujian yang diinginkan. Maka akan dibuat beberapa scenario terhadap pengujian manajemen bandwith *simple queue* dan *queue tree protokol layer 7*. Untuk resume scenario yang akan dilakukan dapat dilihat pada tabel dibawah.

Tabel 3.2 Variabel Pengukuran

No.	Variabel	Tujuan	Metode	Satuan
1.	Speed-limit	Membatasi penggunaan jaringan internet	Dengan menggunakan pembatasan bandwith pada komputer user	kbps

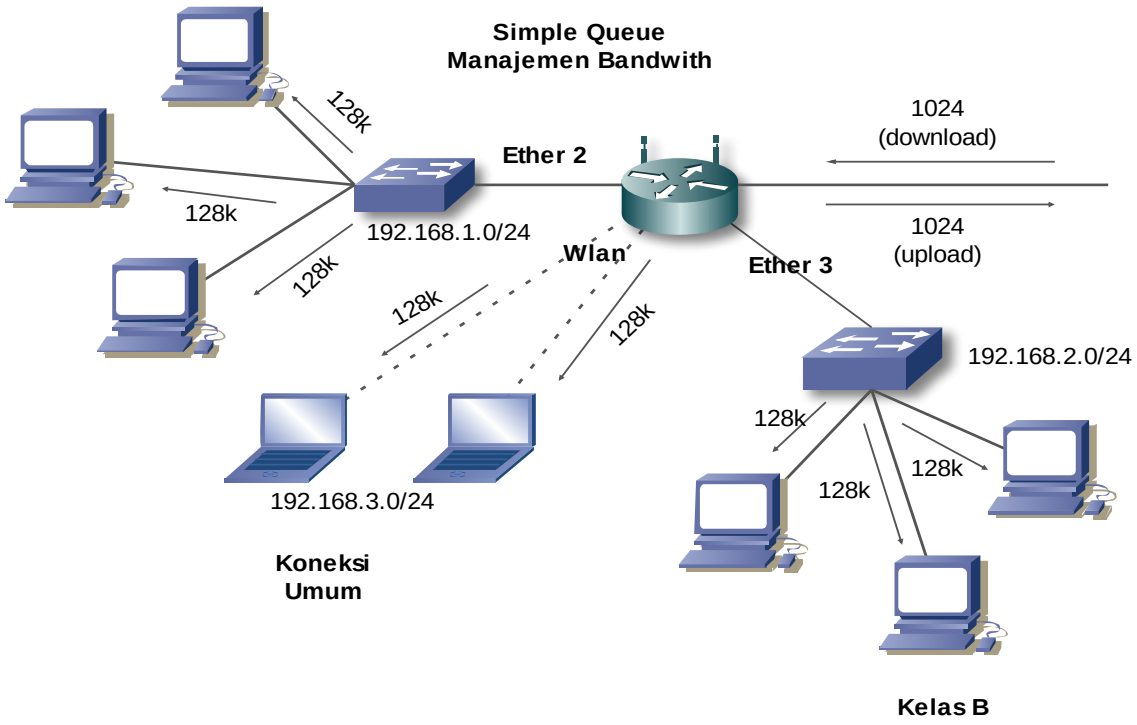
2.	Drop	Membatasi hak akses untuk file-file tertentu	Dengan menggunakan fungsi protokol layer 7 untuk file ekstensi yang ingin dibuang	Gagal atau berhasil
3.	QoS manajemen bandwith	Mengukur besar efisiensi quality of service manajemen bandwith	Dengan membedakan setiap bandwith yang digunakan yang terdiri dari 128 kbps, 256 kbps, 512 kbps, 1024 kbps, dan 2048 kbps	Throughput, delay, packet loss
4.	Resource Memori	Mengukur besar penggunaan beban memori yang ada pada router board	Dengan menggunakan ip graph yang ada pada browser	Persen (%)

Topologi yang akan dibuat pada konfigurasi MikroTik menggunakan dasar topologi yang sudah ada. Beberapa objek yang digunakan pada penelitian ini yaitu 1 router, beberapa klien dan 1 buah server dengan bandwith 2048 kbps baik dari *server* ke router maupun dari router ke *server*.

3.5.1 Pengujian Fungsionalitas Simple Queue dan Queue Tree Layer 7

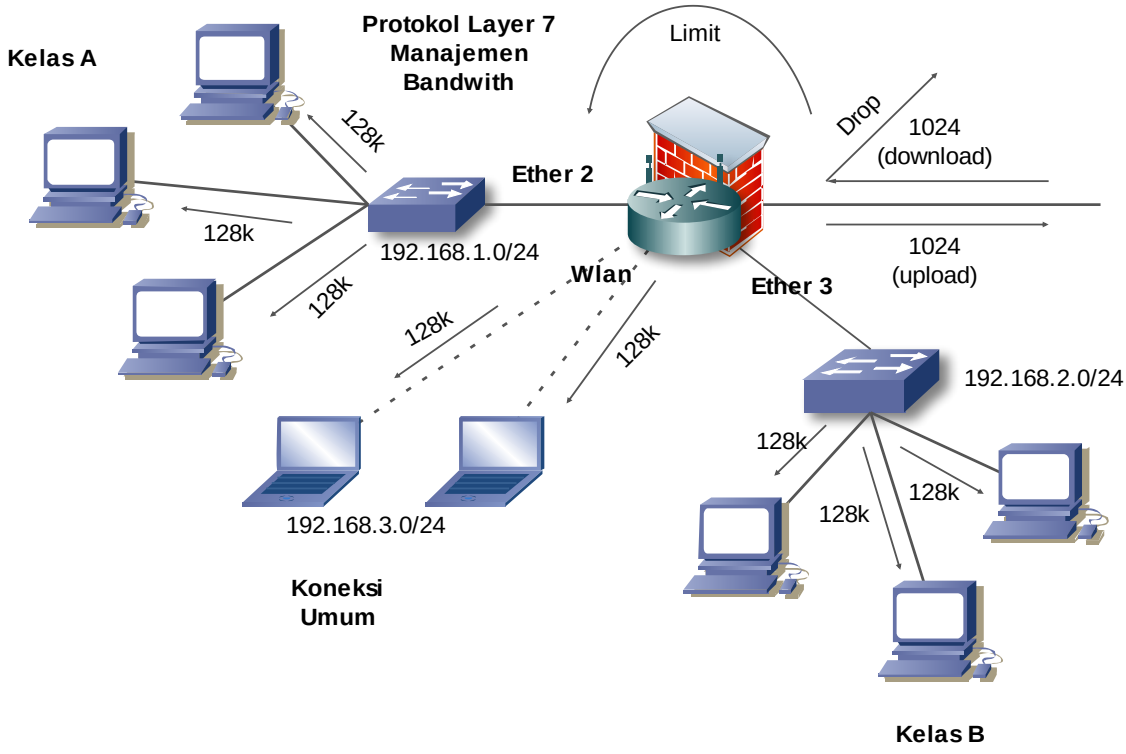
Pada skenario ini diimplementasikan konfigurasi pada router. Semua komputer klien akan mendapatkan bandwith sebesar 128 kbps untuk unggah maupun unduh. Bandwith sebesar 128 kbps merupakan harga CIR, sekaigus MIR. Ini berarti dalam kadaan apapun, baik *traffic* jaringan dalam keadaan sibuk maupun sepi, setiap user akan mendapatkan bandwith sebesar 128 kbps/128 kbps (128 kbps pertama merupakan nilai unggah sedangkan 128 kbps kedua merupakan nilai unduh). Skenario ini juga menggunakan *Destination Address* dalam penggunaan pengaturan pembatasan bandwith. *IP Destination Address* merupakan alamat tujuan yang akan

diberikan terhadap permintaan klien ke *server*. Berikut merupakan gambar skenario yang digunakan untuk manajemen *simple queue* :



Gmbar 3.9 Skenario Simple Queue

Pada skenario ini diimplementasikan konfigurasi pada router. Komputer klien akan mendapatkan bandwith sebesar 128 kbps. Pada skenario *layer 7* memang akan dikonfigurasi seperti konfigurasi yang ada pada *simple queue*. Pada skenario ini nantinya akan dilakukan beberapa konfigurasi tambahan yang sesuai dengan aturan pada *protokol layer 7*. Dalam pengaturan *layer 7*, konten konten akan diblok maupun dilimitasi secara keseluruhan karena akan digunakan *extension*. Router akan digunakan sebagai firewall untuk melakukan *drop* paket. Berikut merupakan skenario penerapan untuk *speed-limit protokol layer 7* :



Gambar 3.10 Skenario Queue Tree Protokol Layer 7

3.5.2 Pengujian Kinerja

3.5.2.1 *Throughput*

Parameter pertama yang diamati untuk mengukur kinerja jaringan menajemen bandwith dalam penelitian ini adalah throughput. *Throughput*, adalah bandwith aktual yang terukur pada suatu ukuran waktu tertentu dalam mentransmisikan berkas. Berbeda dengan bandwith walaupun satuannya sama *kilobit persecond* (kbps), tapi *throughput* lebih menggambarkan bandwith yang sebenarnya pada suatu waktu dan pada kondisi dan jaringan tertentu yang digunakan untuk mengunduh *file* dengan ukuran tertentu. Rumus utnuk menentukan throughput jaringan komputer sebagai berikut :

$$tp = dz / t$$

Keterangan :

tp = throughput

dz = ukuran data yang dikirim

t = time

Untuk mengukur *throughput* pada pengguna manajemen bandwith menggunakan *simple queue* dan *protokol layer 7*, mengunduh beberapa *file* dari *server* dengan ukuran tertentu dengan membatasi bandwith yang digunakan sebesar nilai tertentu dihitung waktu yang dibutuhkan sampai pengunduhan selesai dengan menggunakan *stopwatch*.

3.5.2.2 *Delay* atau *Latency*

Parameter kedua yang diamati untuk mengukur kinerja jaringan manajemen bandwith dalam penelitian ini adalah *delay* atau *latency*. *Latency* adalah waktu yang dibutuhkan untuk sebuah paket yang dikirimkan dari suatu komputer ke komputer yang dituju. *Delay* dalam sebuah proses transmisi paket dalam sebuah jaringan komputer disebabkan karena adanya antrian yang panjang, atau mengambil rute lain untuk menghindari kemacetan pada routing. Untuk mencari delay dalam paket yang ditransmisikan dengan membagi antara panjang paket (satuannya bit) dibagi dengan link bandwith (satuannya bit/s). Untuk mengukur *delay* pada suatu jaringan komputer menggunakan perintah ping yang merupakan salah satu perintah yang dimiliki oleh command prompt sistem operasi windows, dimana time pada hasil perintah ping yang menunjukan delay pada paket yang dikirimkan.

3.5.2.3 *Packet Loss*

Parameter ketiga yang diamati untuk mengukur kinerja jaringan manajemen bandwith dalam penelitian ini adalah *packet loss*.Packet loss adalah persentase paket yang hilang selama mentransmisikan data. Hal ini disebabkan oleh banyak faktor seperti penurunan signal dalam media jaringan, kesalahan perangkat keras jaringan, atau juga radiasi dari lingkungan sekitar. Pada beberapa *network transfer protocol* seperti TCP yang bersifat *connection oriented*, menyediakan pengiriman kembali (*retransmission*) atau pengiriman secara otomatis (*resends*) paket yang hilang selama

proses transmisi walau segmen telah tidak diakui. Walaupun TCP memiliki kelebihan tersebut, jika TCP melakukan *retransmitting* atau *resend*, *throughput* jaringan semakin menurun. Rumus untuk menghitung *packetloss*, adalah sebagai berikut :

Keterangan :

pl = ((pt-pr)/pt) x100%

pl = packet loss

pt = paket yang dikirim

pr = paket yang diterima

3.5.2.4 *Resource Memory*

Parameter keempat yang diamati untuk mengukur kinerja dari masing-masing manajemen bandwith dalam penelitian ini yaitu *Resource Memory*. *Resource memory* merupakan sumber daya dari memori yang membangun kinerja suatu unit (tanyapedia.com. 2013).

BAB IV
HASIL DAN PEMBAHASAN

Bab ini peneliti menampilkan implementasi dan hasil pengujian yang telah dilakukan dilakukan. Untuk mempermudah dalam memahami hasil pengujian bab ini dibagi menjadi beberapa subbab sesuai dengan parameter-parameter pengukuran yang digunakan untuk analisis tentang hasil pengujian.

4.1 Impelementasi Secara Umum

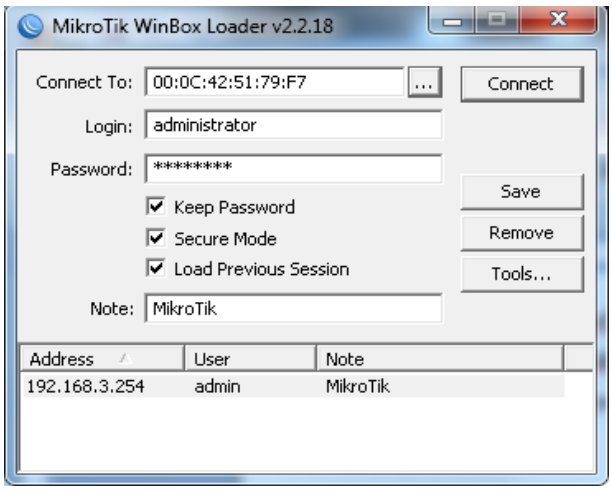
Implementasi sistem merupakan tahap dimana sistem mampu diaplikasikan dalam keadaan yang sesungguhnya. Dari hasil implementasi inilah akan diketahui apakah sistem yang dibuat dapat berjalan dengan baik atau tidak.

4.1.1 Tahapan Implementasi Perangkat Lunak

Segala sesuatu yang berhubungan dengan perangkat lunak diinstall pada mesin dengan metode yang berbeda-beda. Konfigurasi yang akan dilakukan dengan menggunakn *Server* sebagai media utama untuk kebutuhan sistem. Berikut merupakan perangkat-perangkat lunak yang akan digunakan pada *server* maupun komputer yang digunakan sebagai *server* :

1. Konfigurasi MikroTik RouterOS™
    a. Jenis paket MikroTik RouterOS™ yang digunakan dalam konfigurasi adalah MikroTik Router RB493 yang sudah terinstall pada hardware routernya
    b. Konfigurasi yang dilakukan dengan mengatur IP secara *static* dan *dynamic* untuk masing-masing klien. Hal ini digunakan untuk routing dan management bandwith yang dilakukan melalui WinBox *application*.

2. Instalasi WinBox
   a. Jenis paket yang digunakan WinBox *Application* untuk konfigurasi yang berjalan pada sistem operasi windows.
   b. Konfigurasi yang dilakukan secara mendasar dengan routing dan manajemen bandwith pada mesin mikrotik yang sudah ter-install.



Gambar 4.1 Tampilan antarmuka Winbox

Gambar 4.1 diatas merupakan tampilan antar muka winbox yang akan digunakan dalam konfigurasi sistem yang akan dibuat.

3. Konfigurasi Alamat IP

   Pada tahap ini dilakukan konfigurasi alamat IP pada routerboard MikroTik RB493, dalam konfigurasi ini sistem menggunakan empat buah ether dan satu Wlan. Ether1 untuk ISP yaitu Koneksi Speedy, ether2 untuk Kelas A, ether3 untuk Kelas B, Wlan1 untuk Koneksi Umum, ether4 untuk Koneksi Server.

4. Konfigurasi DHCP

   Pada tahapan ini RouterBoard ini akan dioperasikan sebagai DHCP server dengan IP address yang akan diberikan 192.168.1.1/24 untuk Kelas A, 192.168.2.1/24 untuk Kelas B, 192.168.3.1 untuk Koneksi Umum.

5. Konfigurasi Manajemen Bandwith *Simple Queue*

   Pada tahap ini dilakukan konfigurasi sistem yang sesuai dengan skenario perancangan sistem untuk melakukan pembatasan bandwith untuk pengunggahan dan pengunduhan. Konfigurasi dilakukan dengan memberikan *IP Static* untuk masing-masing komputer yang akan dilimitasi.

6. Konfigurasi Manajemen Bandwith *Queue Tree*

   Pada tahap ini dilakukan konfigurasi sistem manajemen bandwith untuk pengunggahan dan pengunduhan seperti konfigurasi *simple queue*, namun pada *queue tree* dilakukan secara *dynamic* untuk komputer user yang ada didalam iTCentrum.

7. Konfigurasi Regelar Expression

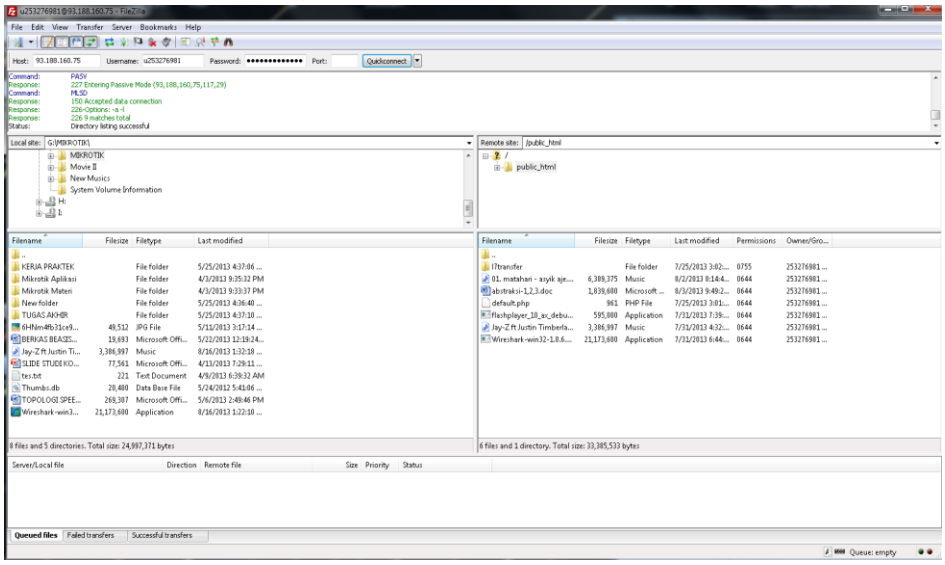
   Pada tahap ini dilakukan dengan menandai pola paket yang nantinya akan dilimitasi maupun dibuang berdasarkan file ekstensinya seperti .mp3, .exe, .doc dan lain-lain.

8. Konfigurasi Firewall

   Pada tahap ini dilakukan konfigurasi mangle terlebih dahulu untuk pangaturan mark paket. Konfigurasi kemudian dilanjutkan dengan membuat filter rule agar mark paket tadi dapat ditentukan apakah akan diterima kemudian dibatasi atau dibuang karena tidak sesuai dengan *rule-rule* yang diterapkan. Setelah itu barulah akan dibuatkan queue tree jumlah bandwithnya untuk mengatur trafik paket pengungguahan dan pengunduhan

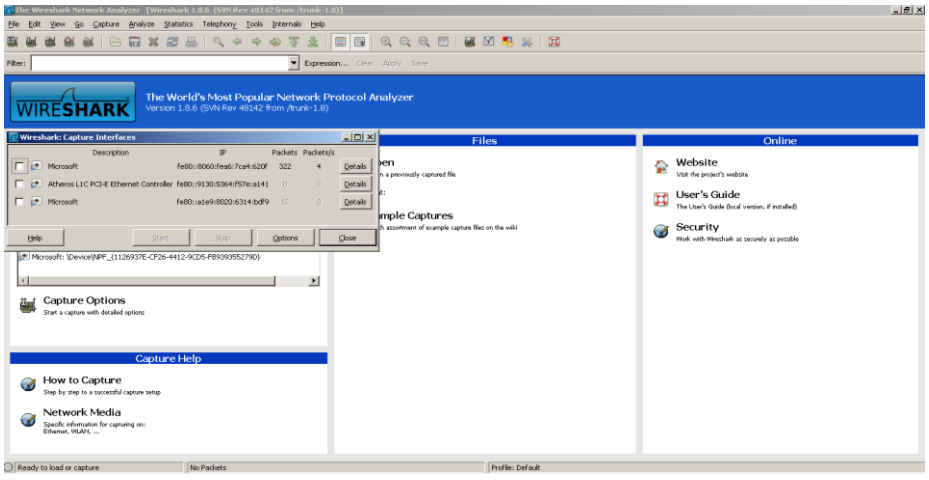
9. Instalasi FileZilla

   a. Perangkat lunak yang digunakan melakukan transfer data
   b. FTP *client*
   c. Menggunakan *server* yang dibuat untuk melakukan proses unggah maupun unduh



Gambar 4.2 Tampilan antarmuka FileZilla

10. Instalasi Wireshark
    a. Menganalisa jaringan
    b. Menangkap paket data atau informasi yang berkeliaran dalam jaringan
    c. Penganalisaan informasi yang dapat dilakukan dengan sniffing, dengan begitu dapat memperolah informasi penting
    d. Membaca data secara langsung dari ethternet dan lain-lain.
    e. Menganalisa transmisi paket data dalam jaringan, proses koneksi, dan transmit data antar komputer dan lain-lain.

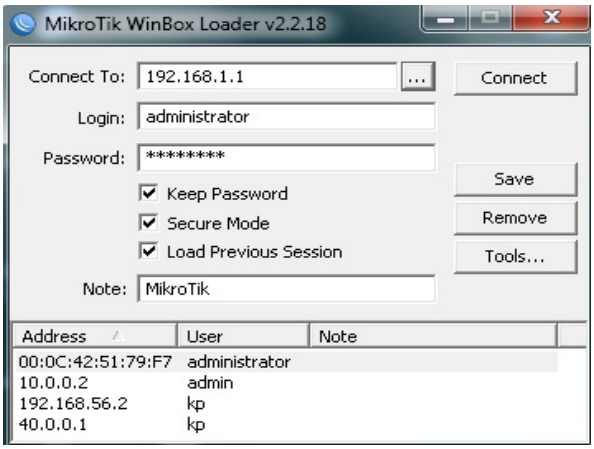


Gambar 4.3 Tampilan antarmuka Wireshark

4.2 Hasil Pengembangan Sistem

4.2.1 Instalasi Winbox

Instalasi winbox, dilakukan dengan meletakkan winbox.exe yang dapat didownload di http://www.mikrotik.com pada komputer yang kemudian bisa langsung dijalankan.Gambar tampilan awal ketika menjalankan winbox.exe. Pada tampilan ini terdapat proses login untuk masuk kedalam konfigurasi MikroTik. Pada isian connect to diisi dengan MAC address dari MikroTik Routerboard. MAC address pada winbox akan muncul secara otomatis apabila RouterBoard sudah terkoneksi. Penggunaan MAC address pada *connect to* dikarenakan belum adanya konfigurasi ether pada RouterBoard. Setelah melakukan konfigurasi pengelamatan IP ether secara otomatis MAC address akan berubah dengan IP Address yang anda konfigurasi. Gambar 4.4 merupakan interface dari RouterBoard MikroTik pada iTCentrum.



Gambar 4.4 Tampilan WinBox

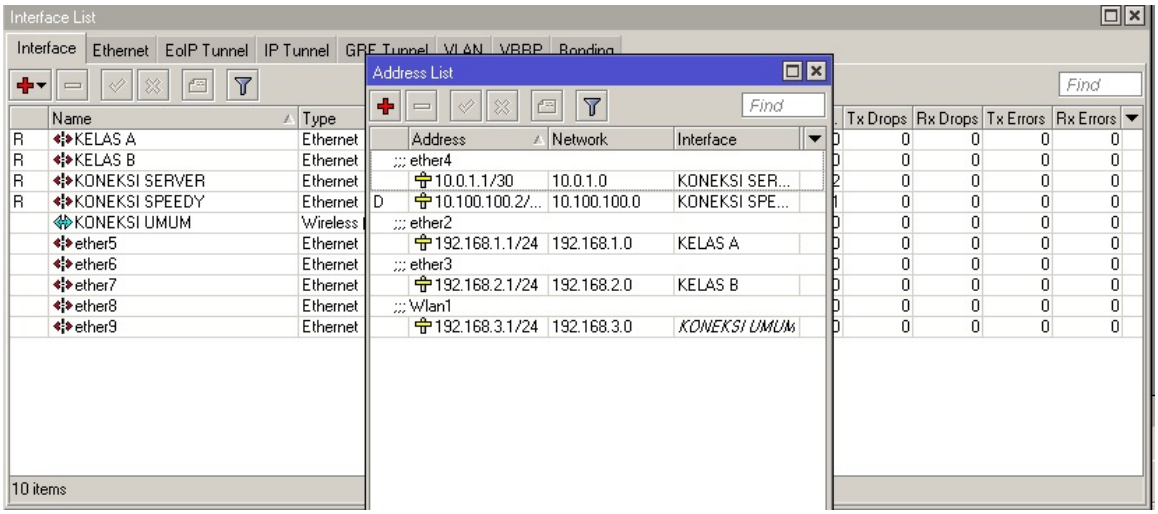
4.2.2 Konfigurasi Alamat IP

RouterBoard MikroTik RB493 mempunyai 9 ether dan 1 Wlan yang tentu dapat dikonfigurasi alamat IP pada setiap ethernya sesuai dengan topologi yang dibangun. Pada mikrotik RB493 penulis menggunakan empat buah ether dan satu buah Wlan. Ether1 digunakan untuk Koneksi Speedy yang merupakan ISP, ehter2 untuk Kelas A, ether3 untuk Kelas B, ether4 untuk Koneksi Server, dan Wlan1 untuk

Koneksi Umum yang merupakan sebuah ether yang disediakan untuk pengguna WiFi.

Gambar 4.4 merupakan tampilan winbox user interface dan menampilkan ether yang tersedia dan yang telah dikonfigurasi. Untuk manampilkan interface list dapat diklik pada menu interface. Gambar 4.4 menampilkan tampilan Address List yang merupakan pemberian alamat IP pada masing-masing ether yang digunakan.pada Address List terlihat ether dan IP yang telah diberikan pada RouterBoard.Berikut merupakan langkah konfigurasi alamat IP yang diberikan untuk masing-masing ether :

1. Koneksi Speedy (ether1) : Address 10.100.100.2/24, Network 10.100.100.0, dan broadcast 10.100.100.255.
2. Kelas A (ether2) : Address 192.168.1.1/24, Network 192.168.1.0, dan Broadcast 192.168.1.255.
3. Kelas B (ether3) : Address 192.168.2.1/24, Network 192.168.2.0, dan Broadcast 192.168.2.255.
4. Koneksi Umum (Wlan1) : Address 192.168.3.1/24, Network 192.168.3.0 dan Broadcast 192.168.3.255



Gambar 4.5 Interface List dan Address List

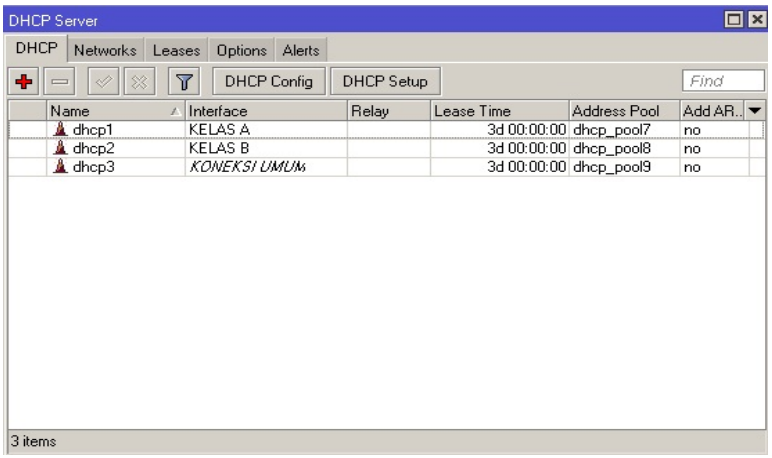
```
[administrator@MikroTik] > interface print
Flags: D - dynamic, X - disabled, R - running, S - slave
 #     NAME                                                TYPE            MTU  L2MTU  MAX-L2MTU
 0  R  KONEKSI SPEEDY                                     ether          1500   1526       1526
 1  R  KELAS A                                            ether          1500   1522       1522
 2  R  KELAS B                                            ether          1500   1522       1522
 3  R  KONEKSI SERVER                                     ether          1500   1522       1522
 4     ether5                                             ether          1500   1522       1522
 5     ether6                                             ether          1500   1522       1522
 6     ether7                                             ether          1500   1522       1522
 7     ether8                                             ether          1500   1522       1522
 8     ether9                                             ether          1500   1522       1522
 9     KONEKSI UMUM                                       wlan           1500   2290
[administrator@MikroTik] >
```

Gambar 4.6 Interface Print

4.2.3 Konfigurasi DHCP Server

Seorang *administrator* jaringan tentu memungkinkan mempunyai banyak aktifitas yang harus dikerjakan. Untuk mengurangi kesibukan tersebut, seorang admin dapat menyediakan *server* DHCP agar tidak perlu lagi member IP address *client* secara manual untuk setiap komputer.

DHCP (*Dynamic Host Configuration Protocol*) adalah suatu protocol yang dapat mendistribusikan IP address, default gateway, DNS server, dan WINS server (untuk client windows) kepada client dalam sebuah jaringan secara otomatis.Tampilan DHCP server dari kelas-kelas yang telah dibuat dapat dilihat pada gambar 4.7.



Gambar 4.7 DHCP server

Pada gambar 4.7 terdapat 3 buah DHCP server yang telah dibuat. Masing-masing DHCP server dibuat untuk kebutuhan kelas-kelas yang ada di iTCentrum. Hal

ini juga dikarenakan perbedaan IP yang ada pada masing-masing kelas.Untuk mengkonfigurasi DHCP server dapat menggunakan DHCP setup yang ada pada menu (IP – DHCP Server – DHCP Setup). Dalam DHCP setup akan dikonfigurasi ether mana yang akan diberikan IP address, gateway, serta dns server yang akan digunakan.

Setelah DHCP server berhasil dikonfigurasi maka secara otomatis komputer user dari masing-masing kelas seperti Kelas A, Kelas B, dan Koneksi Umum akan mendapatkan IP untuk setiap komputer yang berada dibawahnya.

4.2.4 Konfigurasi Skenario Manajemen Bandwith *Simple Queue*

Seperti yang telah penulis bahas sistem menggunakan dua buah skenario untuk perekayasaan sistem manajemen bandwith yang ada pada jaringan iTCentrum. Pertama-tama akan dibuat manajemen bandwith dengan menggunakan queue simple.

4.2.4.1 *Simple Queue* Kelas A

Konfigurasi manajemen yang pertama dibuat yaitu *simple queue* untuk Kelas A.

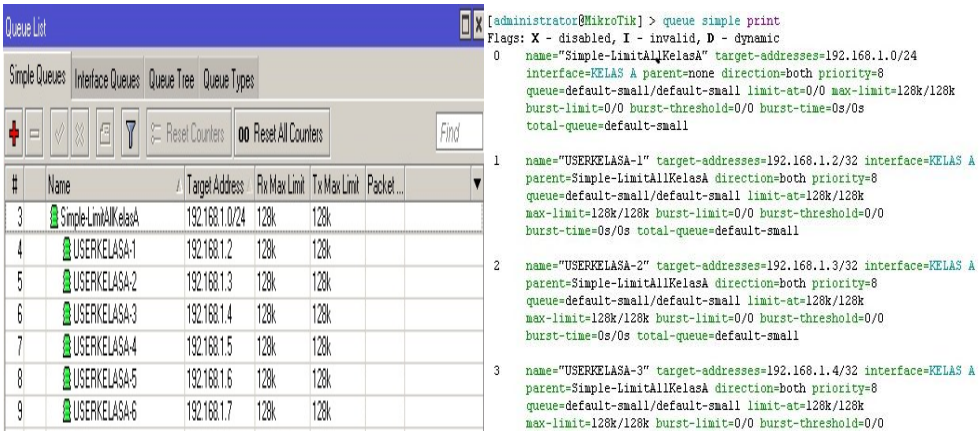
```
[administrator@MikroTik] > queue simple add name=Simple-LimitAllKelasA target-addresses=192.168.1.0/24
 interface="KELAS A" max-limit=128000/128000
[administrator@MikroTik] > queue simple add name=USERKELASA-1 target-addresses=192.168.1.2 interface="
KELAS A" max-limit=128000/128000 limit-at=128000/128000 parent=Simple-LimitAllKelasA
[administrator@MikroTik] > queue simple add name=USERKELASA-2 target-addresses=192.168.1.3 interface="
KELAS A" max-limit=128000/128000 limit-at=128000/128000 parent=Simple-LimitAllKelasA
[administrator@MikroTik] > queue simple add name=USERKELASA-3 target-addresses=192.168.1.4 interface="
KELAS A" max-limit=128000/128000 limit-at=128000/128000 parent=Simple-LimitAllKelasA
[administrator@MikroTik] > queue simple add name=USERKELASA-4 target-addresses=192.168.1.5 interface="
KELAS A" max-limit=128000/128000 limit-at=128000/128000 parent=Simple-LimitAllKelasA
[administrator@MikroTik] > queue simple add name=USERKELASA-5 target-addresses=192.168.1.6 interface="
KELAS A" max-limit=128000/128000 limit-at=128000/128000 parent=Simple-LimitAllKelasA
[administrator@MikroTik] > queue simple add name=USERKELASA-6 target-addresses=192.168.1.7 interface="
KELAS A" max-limit=128000/128000 limit-at=128000/128000 parent=Simple-LimitAllKelasA
[administrator@MikroTik] >
```

Gambar 4.8 Konfigurasi *Simple Queue* Kelas A

Pada Gambar 4.8 dapat dilihat merupakan konfigurasi simple queue Kelas A untuk jaringan iTCentrum. Disini admin menggunakan terminal yang ada pada MikroTik untuk mengkonfigurasi semua keperluan yang menyangkut dengan sistem yang dibangun. Simple queue yang dibuat diberi nama Simple-LimitAllKelasA dengan target-addresses=192.168.1.0/24, interface=”Kelas A” (ether yang digunakan)

dengan memberika pembatasan bandwith sebesar 128 kbps (upload) dan 128 kbps (unduh). Simple-LimitAllKelas A merupakan induk dari *simple queue* untuk Kelas A. Setelah itu admin membuat *simple queue* untuk komputer yang akan diberikan *speed-limit.*

Penulis menggunakan enam buah komputer Kelas A sebagai rekayasa sistem yang dibuat. *Simple queue* yang dibuat yaitu dengan nama USERKELASA-1, target-address 192.168.1.2, sampai dengan USERKELASA-6, target-addresses 192.168.1.7 dengan interface yang sama, untuk unggah dan unduh masing-masing sebesar 128Kb, serta menggunakan parent=Simple-LimitAllKelasA agar aktifitas diluar IP yang dikonfigurasi juga dapat dikontrol oleh Simple queue induk.



Gambar 4.9 Inteface *Simple Queue* Kelas A dan Terminal Print

4.2.4.2 Simple Queue Kelas B

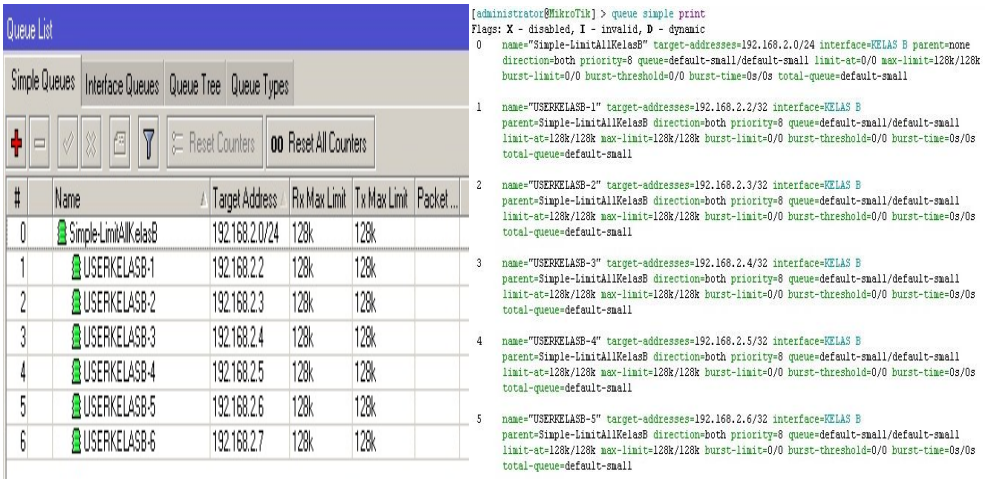
Konfigurasi Menajemen Bandwith yang kedua adalah *simple queue* Kelas B.

```
[administrator@MikroTik] > queue simple add name=Simple-LimitAllKelasB target-addresses=192.168.2.0/24
 interface="KELAS B" max-limit=128000/128000
[administrator@MikroTik] > queue simple add name=USERKELASB-1 target-addresses=192.168.2.2 interface="
KELAS B" max-limit=128000/128000 limit-at=128000/128000 parent=Simple-LimitAllKelasB
[administrator@MikroTik] > queue simple add name=USERKELASB-2 target-addresses=192.168.2.3 interface="
KELAS B" max-limit=128000/128000 limit-at=128000/128000 parent=Simple-LimitAllKelasB
[administrator@MikroTik] > queue simple add name=USERKELASB-3 target-addresses=192.168.2.4 interface="
KELAS B" max-limit=128000/128000 limit-at=128000/128000 parent=Simple-LimitAllKelasB
[administrator@MikroTik] > queue simple add name=USERKELASB-4 target-addresses=192.168.2.5 interface="
KELAS B" max-limit=128000/128000 limit-at=128000/128000 parent=Simple-LimitAllKelasB
[administrator@MikroTik] > queue simple add name=USERKELASB-5 target-addresses=192.168.2.6 interface="
KELAS B" max-limit=128000/128000 limit-at=128000/128000 parent=Simple-LimitAllKelasB
[administrator@MikroTik] > queue simple add name=USERKELASB-6 target-addresses=192.168.2.7 interface="
KELAS B" max-limit=128000/128000 limit-at=128000/128000 parent=Simple-LimitAllKelasB
[administrator@MikroTik] >
```

Gambar 4.10 Konfigurasi *Simple Queue* Kelas B

Gambar 4.10 merupakan *simple queue* kelas B untuk jaringan iTCentrum. Untuk langkah konfigurasi yang dilakukan hampir sama dengan yang dilakukan untuk kelas A hanya saja penggunaan konfigurasi IP yang tentunya berbeda. Untuk Induk simple queue Kelas B digunakan nama Simple-LimitKelasB target-addresses=192.168.2.0/24, interface=”Kelas B” dan max-limit=128Kb/128Kb untuk masing-masing unggah dan unduh.

Kelas B juga menggunakan enah buah komputer user akan dilimitasi. Simple queue yang dibuat dengan nama USERKELASB-1, target-addresses=192.168.2.2 sampai dengan USERKELASB-6, target-addresses=192.168.2.7 interface=”Kelas B” max-limit=128Kb/128Kb, limit-at=128Kb/128Kb parent=Simple-LimitKelasB.



Gambar 4.11 Interface *Simple Queue* Kelas B dan Terminal Print

4.2.4.3 Simple Queue Koneksi Umum

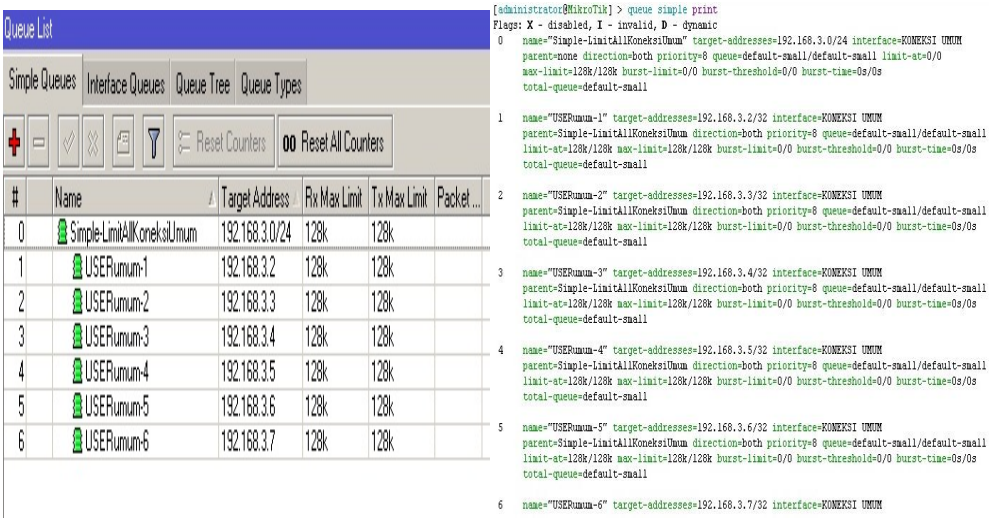
Konfigurasi manajemen bandwith yang terakhir adalah *simple queue* untuk Konkesi Umum yang merupakan Koneksi Menggunakan Wifi.

```
[administrator@MikroTik] > queue simple add name=Simple-LimitAllKoneksiUmum target-addresses=192.168.3
.0/24 interface="KONEKSI UMUM" max-limit=128000/128000
[administrator@MikroTik] > queue simple add name=USERumum-1 target-addresses=192.168.3.2 interface="KO
NEKSI UMUM" max-limit=128000/128000 limit-at=128000/128000 parent=Simple-LimitAllKoneksiUmum
[administrator@MikroTik] > queue simple add name=USERumum-2 target-addresses=192.168.3.3 interface="KO
NEKSI UMUM" max-limit=128000/128000 limit-at=128000/128000 parent=Simple-LimitAllKoneksiUmum
[administrator@MikroTik] > queue simple add name=USERumum-3 target-addresses=192.168.3.4 interface="KO
NEKSI UMUM" max-limit=128000/128000 limit-at=128000/128000 parent=Simple-LimitAllKoneksiUmum
[administrator@MikroTik] > queue simple add name=USERumum-4 target-addresses=192.168.3.5 interface="KO
NEKSI UMUM" max-limit=128000/128000 limit-at=128000/128000 parent=Simple-LimitAllKoneksiUmum
[administrator@MikroTik] > queue simple add name=USERumum-5 target-addresses=192.168.3.6 interface="KO
NEKSI UMUM" max-limit=128000/128000 limit-at=128000/128000 parent=Simple-LimitAllKoneksiUmum
[administrator@MikroTik] > queue simple add name=USERumum-6 target-addresses=192.168.3.7 interface="KO
NEKSI UMUM" max-limit=128000/128000 limit-at=128000/128000 parent=Simple-LimitAllKoneksiUmum
[administrator@MikroTik] >
```

Gambar 4.12 Konfigurasi *Simple Queue* Koneksi Umum

Gambar 4.12 merupakan *simple queue* dari Wlan1 Koneksi Umum. Untuk induk *simple queue* Koneksi Umum diberi nama Simple-LimitAllKoneksiUmum target-addresses=192.168.3.0/24 interface=”Koneksi Umum” (ether Wlan1) dan max-limit=128kb/128kb masing-masing unggah dan unduh.

Kelas B juga menggunakan enah buah komputer user yang digunakan untuk dilimitasi. *Simple queue* user diberi nama Userumum-1, target-addresses 192.168.3.2 sampai dengan Userumum-6, target-addresses 192.168.3.7, dengan interface=”KONEKSI UMUM”, max-limit=128Kb/128Kb serta parent=Simple-LimitAllKoneksiUmum.



Gambar 4.13 Interface *Simple Queue* Koneksi Umum dan Terminal Print

Langkah-langkah di atas merupakan konfigurasi awal dari skenario simple queue yang dibuat didalam rancanang sistem jaringan iTCentrum, manajemen bandwith yang diterapkan pada sistem akan diuji lagi untuk mendapatkan hasil penerapan sistem yang paling baik untuk jaringan iTCentrum.

4.2.5 Konfigurasi Skenario Manajemen Bandwith *Queue Tree* Menggunakan *Protokol Layer 7*

Konfigurasi ini bertujuan untuk mengatur besar bandwith yang diterima setiap klien. Konspenya hampir sama seperti *simple queue* namun pada konfigurasi ini ditambahkan Firewall yang berfungsi sebagai *rule* yang akan memeriksa paket yang akan diberikan *speed-limit* atau dibuang. Konfigurasi manajemen bandiwith menggunakan metode-metode yang disesuaikan dengan *queue tree*.

4.2.5.1 Pembuatan Firewall Regular Expression

Regular Expression Layer 7 digunakan pencari dan pengganti pola *text* dan juga sebagai penanda untuk paket file ekstensi yang terdapat pada *Layer 7*. Pada konfigurasi ini file ekstensi yang akan dimanajemen bandwith dideskripsikan dan yang akan diperiksa oleh *rule-rule* selanjutnya. Gambar 4.14 adalah konfigurasi menambah *Regular Expression* untuk queue tree.

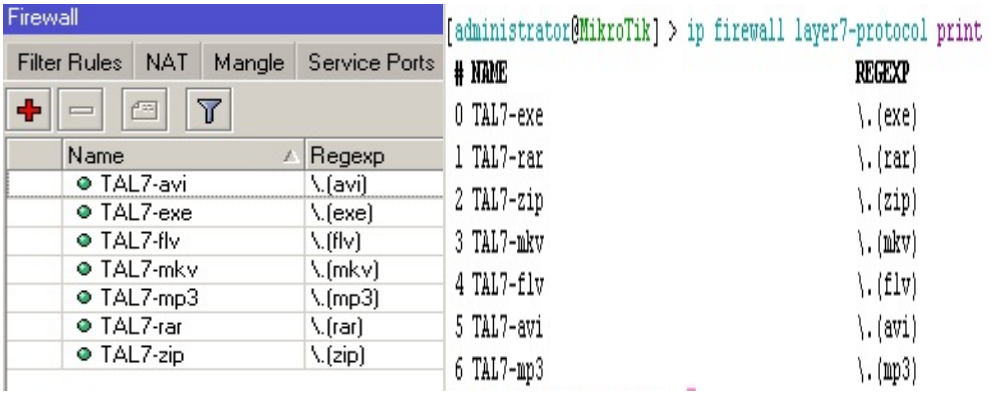
```
[administrator@MikroTik] > /ip firewall layer7-protocol add name="TAL7-exe" regexp="\\
.(exe)"
[administrator@MikroTik] > /ip firewall layer7-protocol add name="TAL7-rar" regexp="\\
.(rar)"
[administrator@MikroTik] > /ip firewall layer7-protocol add name="TAL7-zip" regexp="\\
.(zip)"
[administrator@MikroTik] > /ip firewall layer7-protocol add name="TAL7-mkv" regexp="\\
.(mkv)"
[administrator@MikroTik] > /ip firewall layer7-protocol add name="TAL7-flv" regexp="\\
.(flv)"
[administrator@MikroTik] > /ip firewall layer7-protocol add name="TAL7-avi" regexp="\\
.(avi)"
[administrator@MikroTik] > /ip firewall layer7-protocol add name="TAL7-mp3" regexp="\\
.(mp3)"
[administrator@MikroTik] >
```

Gambar 4.14 Konfigurasi Regular Expression Protocol Layer 7

Pada gambar 4.14 dapat dilihat konfigruasi *IP firewall Layer 7 Protocol*. Menggunakan terminal yang ada pada MikroTik admin mendeskripsikan RegExp

file-file extensi yang akan dimanajemen bandwith oleh queue tree. File extensi diberi nama TAL7-(file extensi) dan regexp dari file ekstensi tersebut. Regexp=\\.(msi) salah satu contoh dari file ekstensi yang dideskripsikan di dalam *protokol layer 7*.

Pada gambar diatas penulis memperlihatkan semua regexp untuk file-file ekstensi yang terdapat dalam *protokol layer 7* sistem iTCentrum. Penulisan regexp sesuai dengan aturan yang ada pada *layer 7*, hal ini dikarenakan apabila tidak sesuai dengan penulisan yang ditetapkan oleh MikroTik maka pola *layer 7* tidak dapat ditangkap oleh RouterBoard MikroTik. Pada sistem yang diterapkan admin mengambil sampel beberapa *file extension* saja yang akan diuji kinerjanya apakah dapat diterapkan atau tidak.



Gambar 4.15 Interface Firewall Protocols Layer 7 dan Terminal Print

4.2.5.2 Konfigurasi *Firewall Filter Rules*

Fitur filter pada firewall digunakan untuk menentukan apakah suatu paket data dapat masuk atau tidak ke dalam sistem router itu sendiri. Paket data yang akanyang akan ditangani fitur filter ini adalah paket yang ditunjukan pada salah satu interface router. Pada sistem yang dibangun digunakan *Chain Forward* untuk menangani paket data yang akan melintasi router.

Chain Forward akan menangani paket data yang melintasi router, baik paket data dari jaringan lokal yang ingin ke internet, maupun sebaliknya. Pada konfigurasi diterapkan beberapa *action* untuk paket yang masuk kedalam router. Pada sistem yang dibuat diterapkan *action accept* dan *action drop*. *Action accept* digunakan untuk memeriksa paket yang diterima tanpa melanjutkan pembacaan aturan selanjutnya.

Sedangkan *action drop* digunakan untuk menolak paket secara diam-diam (tanpa mengirim balasan penolakan dari paket ICMP)

```
[administrator@MikroTik] > /ip firewall filter add chain=forward layer7-protocol="TAL7-exe"
 action=dropaction=drop
[administrator@MikroTik] > /ip firewall filter add chain=forward layer7-protocol="TAL7-rar"
 action=dropaction=drop
[administrator@MikroTik] > /ip firewall filter add chain=forward layer7-protocol="TAL7-zip"
 action=dropaction=drop
[administrator@MikroTik] > /ip firewall filter add chain=forward layer7-protocol="TAL7-mkv"
 action=dropaction=drop
[administrator@MikroTik] > /ip firewall filter add chain=forward layer7-protocol="TAL7-flv"
 action=dropaction=drop
[administrator@MikroTik] > /ip firewall filter add chain=forward layer7-protocol="TAL7-avi"
 action=dropaction=drop
[administrator@MikroTik] > /ip firewall filter add chain=forward layer7-protocol="TAL7-mp3"
 action=dropaction=drop
[administrator@MikroTik] >
```

Gambar 4.16 Konfigurasi *Firewall Filter Rules*

Pada gambar 4.16 diterapkan beberapa firewall filter rules untuk file-file ekstensi yang tadi telah dideskripsikan di *Layer 7 Protocol*. File-file ekstensi yang akan difilter yaitu exe, rar, zip, mkv, flv, avi, dan mp3 dengan menggunakan chain=forward dan action=drop. Filter rules yang digunakan memang untuk melakukan proses pembuangan paket data yang tidak diinginkan masuk kedalam router. Admin dapat membuat apakah paket-paket file ekstensi dapat masuk atau tidak kedalam router. *Action accept* digunakan apabila paket file ekstensi dipersilahkan untuk masuk kedalam router yang di unduh oleh user namun paket file tersebut tetap akan diproses dalam manajemen bandwith yaitu queue tree agar mendapatkan speed-limit pengunduhan.

4.2.5.3 Konfigurasi *Mangle*

Mangle adalah aturan pada pembatasan bandwith. Pada *queue tree* menggunakan satu buah *mark*, yaitu *mark packet*. *Mark* digunakan untuk menandai jalur paket setiap klien. Setelah membuat *Protokol Layer 7* dan *Firewall Filter Rule* kemudian dilakukan konfigurasi *Mangle*. File-file ekstensi *protokol Layer 7* tadi ditandai dan penandaan ini dapat diterapkan untuk fitur-fitur mikrotik lainnya. Fitur yang akan digunakan nantinya yaitu *queue tree* dimana apabila paket-paket file ekstensi tadi diterima maka akan dilanjutkan ke proses *queue tree*.

Konfigurasi *firewall mangle* dapat dilihat pada gambar 4.17. *Chain* yang digunakan adalah chain=forward, protocol=tcp, layer7-protocol=nama file ekstensi, dan dengan menggunakan new-mark-packet=download yang nanti akan dideskripsikan di *queue tree*.

```
[administrator@MikroTik] > /ip firewall mangle add chain=forward protocol=tcp layer7-protoc
ol=TAL7-exe action=mark-packet new-packet-mark=download passthrough=nosthrough=n
[administrator@MikroTik] > /ip firewall mangle add chain=forward protocol=tcp layer7-protoc
ol=TAL7-rar action=mark-packet new-packet-mark=download passthrough=nosthrough=n
[administrator@MikroTik] > /ip firewall mangle add chain=forward protocol=tcp layer7-protoc
ol=TAL7-zip action=mark-packet new-packet-mark=download passthrough=nosthrough=n
[administrator@MikroTik] > /ip firewall mangle add chain=forward protocol=tcp layer7-protoc
ol=TAL7-mkv action=mark-packet new-packet-mark=download passthrough=nosthrough=n
[administrator@MikroTik] > /ip firewall mangle add chain=forward protocol=tcp layer7-protoc
ol=TAL7-flv action=mark-packet new-packet-mark=download passthrough=nosthrough=n
[administrator@MikroTik] > /ip firewall mangle add chain=forward protocol=tcp layer7-protoc
ol=TAL7-avi action=mark-packet new-packet-mark=download passthrough=nosthrough=n
[administrator@MikroTik] > /ip firewall mangle add chain=forward protocol=tcp layer7-protoc
ol=TAL7-mp3 action=mark-packet new-packet-mark=download passthrough=nosthrough=n
[administrator@MikroTik] >
```

Gambar 4.17 Konfigurasi Firewall Mangle

4.2.5.4 Konfigurasi Manajemen Bandwith *Queue Tree*

Langkah terakhir dari skenario manajemen bandwith *queue tree* yaitu mebuat konfigurasi *queue tree*. Konfigurasi ini mengatur bandwith unggah dan unduh dari konfigurasi-konfigurasi sebelumnya dari skenario ini. *Queue tree* merupakan tahap paling akhir setelah paket yang masuk diperiksa melalui *rule-rule Filter* dan *Mangle* yang kemudian dilanjutkan ke aturan ini.

Gambar 4.18 merupakan konfigurasi *queue tree*. Dalam aturan ini paket akan mendapatkan bandwith sebesar 128 kbps, 256 kbps, 512 kbps, 1024 kbps, dan 2048 kbps sesuai dengan pengujian data yang akan dilakukan. *Queue tree* diberi nama total-download, parent=global-out, packet-mark=download diambil dari new mark packet dari mangle yang telah dibuat, serta limit-at=128Kb.

```
[administrator@MikroTik] > /queue tree add name="total-download" parent=global-out packet-m
ark=download limit-at=128k queue=default priority=8 max-limit=384k burst-limit=0 burst-thre
shold=0 burst-time=0s
[administrator@MikroTik] >
```

Gambar 4.18 Konfigurasi *Qeueu Tree*

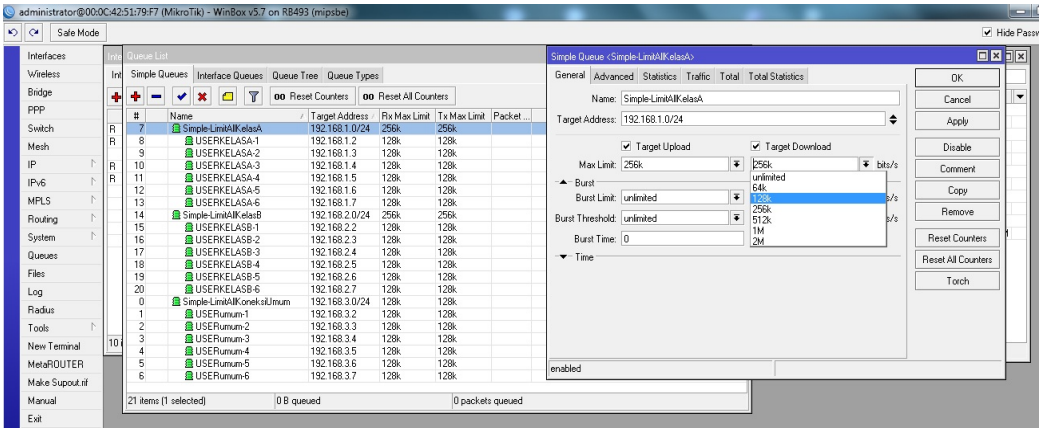
4.3 Pengujian Implementasi

4.3.1 Pengujian Skenario *Simple Queue*

4.3.1.1 Manajemen Bandwith

Untuk pengujian skenario ini dilakukan beberapa tahapan yang mendukung untuk pengujian skenario manajemen bandwith *simple queue*. Pada tahapan ini dilakukan manajemen bandwith dari komputer *client* yang akan diuji. Berdasarkan skenario yang telah dirancang, penulis akan melakukan pembatasan bandwith untuk setiap kelas-kelas yang adap ada iTCentrum. Pembatasan bandwith ini tentu akan mempengaruhi bandwith yang didapat untuk unggah dan unduh dari FileZilla yang akan digunakan untuk pengujian data. Untuk melakukan pengaturan Bandiwith simple yaitu dengan masuk kedalam MikroTik RouterBoard dengan menggunakan WinBox.

Untuk masuk kedalam *queue* yang ada pada MikroTik yaitu dengan menggunakan queue pada menu. Pada Gambar 4.19 dapat dilihat merupakan konfigurasi dari *simple queue* Kelas A, Kelas B, dan Koneksi Umum yang telah dikonfigurasi terlebih dahulu. Karena menggunakan *simple queue* user manajemen bandwith harus dibuat secara static untuk setiap user. Sistem yang dirancang untuk *simple queue* menggunakan 5 buah klien yang sebagai sampel. Untuk pengujian sendiri akan diambil salah satu klien dari masing-masing kelas dengan melakukan pembatasan bandwith dari 128 kbps, 256 kbps, 512 kbps, 1024 kbps dan 2048 kbps.

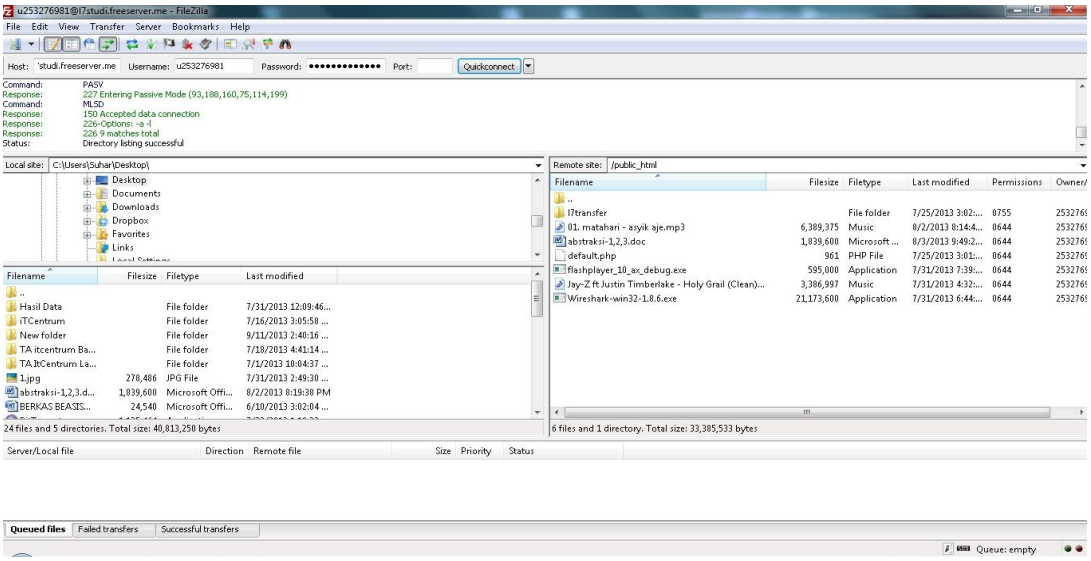


Gambar 4.19 Bandiwith Manajemen *Simple Queue*

4.3.1.2 FileZilla Remote

Seperti yang sudah diuraikan untuk perangkat-perangkat lunak pendukung yang digunakan untuk pengujian implementasi rancangan sistem yang dibuat, pertama-tama dengan perangkat lunak FTP Client FileZilla.Untuk menggunakan FileZilla terlebih dahulu harus mempunyai akun di Webserver Hosting.Untuk akun sendiri dapat menggunakan Webserver yang bisa didaftarkan secara gratis.Disini penulis menggunakan Webserver 2FreeHosting.com yang mendukung FTP Client Filezilla. Seperti yang sudah diketahui FTP Cilent FileZilla memudahkan untuk proses pengirimin unggah maupun unduh.

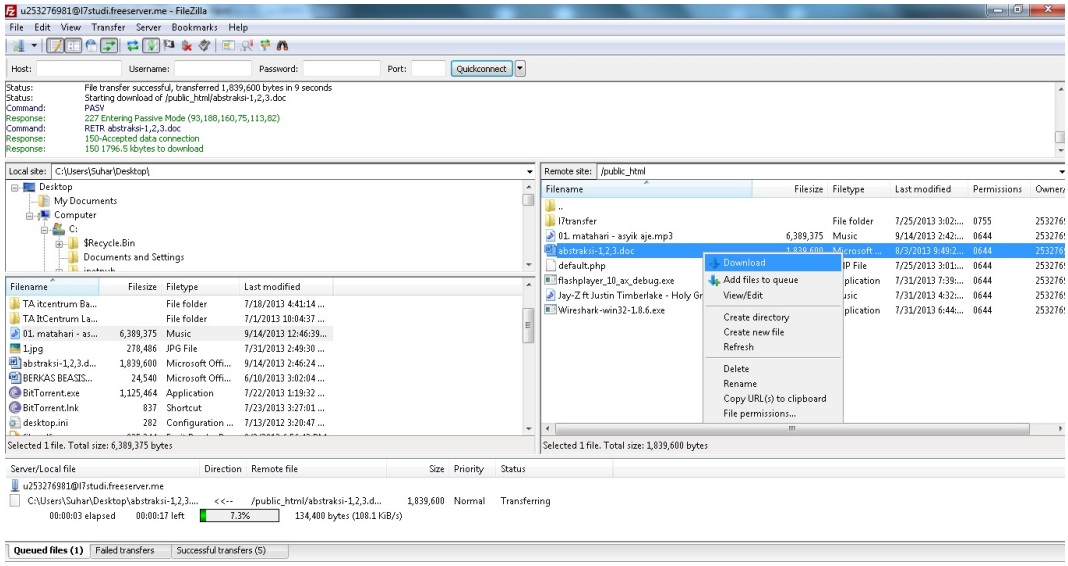
Gambar 4.20 merupakan tampilan dari FTP Client FileZilla dengan status sudah login dengan akun yang telah didapatkan dari Webserver yang sudah kita daftarkan tadi.



Gambar 4.20 Login FTP Cilent FileZilla

Untuk melakukan proses pengunduhan paket data yang ada. Terlebih dahulu melakukan proses mengunggah file-file yang akan diujikan kedalam Webserver. Untuk mengunggah file yang akan diteliti anda dapat menggukan kolom sebelah kiri yang merupakan *space* dari komputer anda sedangkan kolom sebelah kanan merupakan hasil proses mengunggah file-file tersebut. Gambar 4.21 merupakan proses melakukan pengunduhan file pada aplikasi FileZilla. Keterangan proses

pengunduhan dapat dilihat pada kolom yang paling bawah. Pada keterangan tersebut menunjukan waktu, kecepatan dari proses pengunduhan yang dilakukan. Kecepatan pengunduhan sendiri telah diatur dalam pembatasan bandiwith yang diberikan oleh *simple queue* dan *queue tree*.

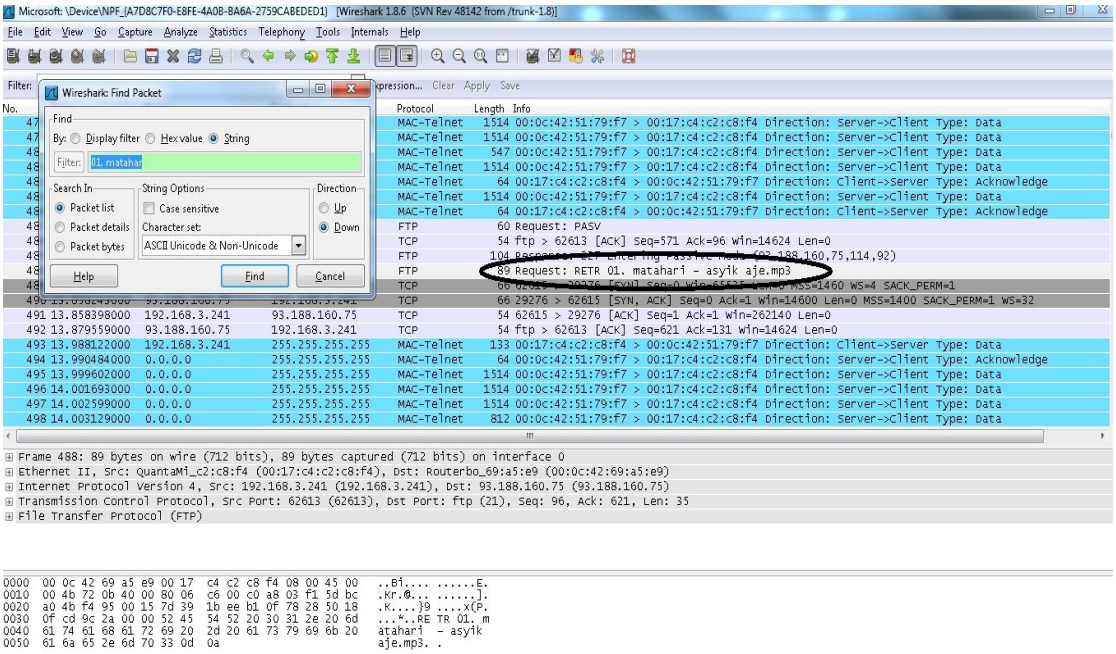


Gambar 4.21 Proses Pengunduhan Peket Data

4.3.1.3 WireShark Analisis Paket Data

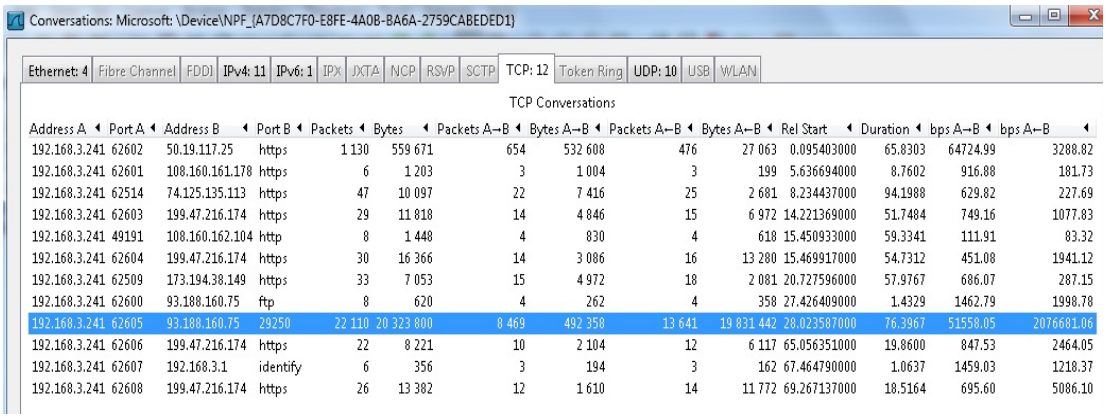
Tahapan ini dilakukan analisis data dari paket yang sedang dalam proses pengujian. Perangkat lunak WireShark dapat melihat semua aktivitas yang ada pada komputer yang terkoneksi dengan jaringan yang ada. Pada perangkat lunak WireShark penulis menggunakannya untuk menemenukan paket data yang dikirim, trafik paket data, maupun proses ping yang dilakukan oleh command prompt.

Proses membatasi bandwith user menggunakan MikroTik, melakukan pengunduhan paket data menggunakan FileZilla, kemudian melakukan pengamatan paket data yang telah dikirimkan seperti *find* paket, *conversation*, dan pengamatan ping dari command prompt. Gambar 4.22 merupakan proses pencarian paket data yang diunduh menggunakan perangkat lunak FileZilla.



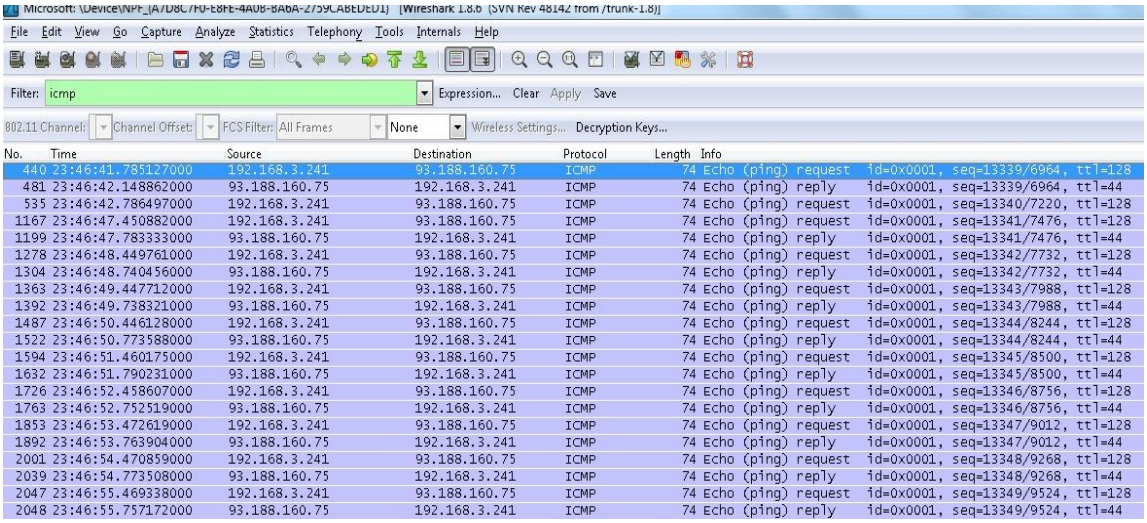
Gambar 4.22 *Find* Paket

Gambar 4.23 merupakan tampilan dari *conversation* WireShark yang sedang berjalan didalam jaringan tersebut. *Conversation* berguna untuk mengetahui proses dari aktifitas yang terjadi didalam jaringan itu sendiri, baik dari ehternet yang digunakan, IPv4, IPv6, TCP, maupun UDP. Karena proses pengiriman paket data terjadi pada TCP maka dapat dilihat pada kolom TCP.



Gambar 4.23 *Conversation* Wireshark

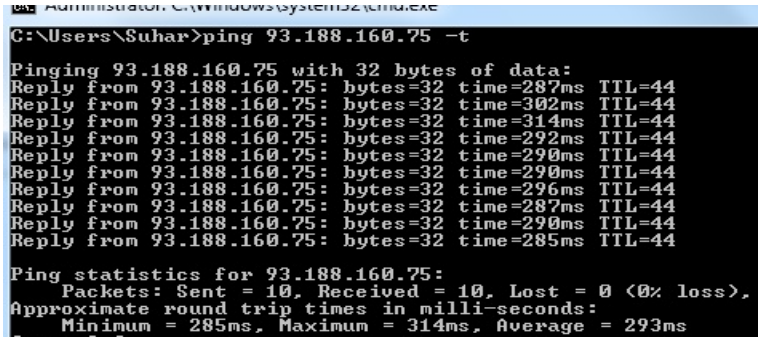
Gambar 4.24 manampilkan tampilan WireShark dengan filter ICMP untuk melakukan perhitungan *delay* dari paket data yang dikirimkan. Kolom *time* merupakan informasi dari waktu pengiriman data, *source* merupakan alamat asal yaitu IP komputer, *Destination* merupakan alamat tujuan yaitu *server*, Protocol ICMP merupakan protokol dari ping Command Prompt, sedangkan info berisi *request* dan *replay* dari *source* dan *destination*.



Gambar 4.24 Perhitungan *Delay*

4.3.1.4 Ping Menggunakan Command Prompt

Tahapan ini merupakan tahapan terakhir dari pengujian implementasi manajemen bandwith untuk mengetahui kinerja dari sistem yang dirancang. Setelah tahapan pengaturan bandiwith, proses pengunduhan paket data, wireshark analisis, maka proses ping dilakukan untuk menghitung packet loss pada paket sekaligus menjadi capture untuk menghitung delay pada WireShark. Untuk menghitung delay dan packet loss penulis menggunakan ping ke 93.188.160. 75 yaitu server webhosting sebagai patokan dasar perhitungan. Gambar 4.25 merupakan hasil ping dari server webhosting dengan ping normal.

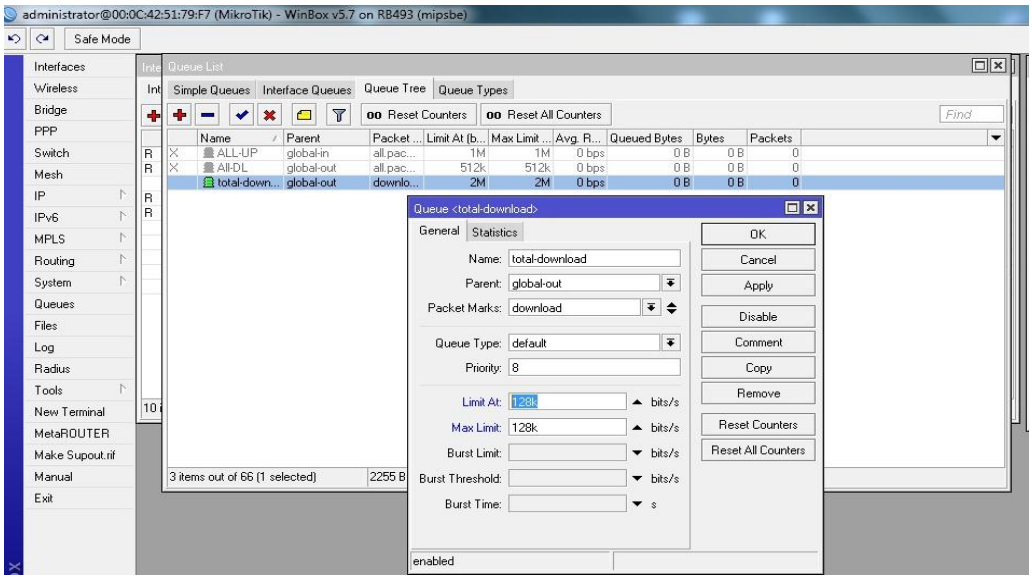


Gambar 4.25 Ping Perhitungan Packet Loss

4.3.2 Pengujian Skenario Firewall Protocol Layer 7 Queue Tree

4.3.2.1 Manajemen Bandwith Queue Tree

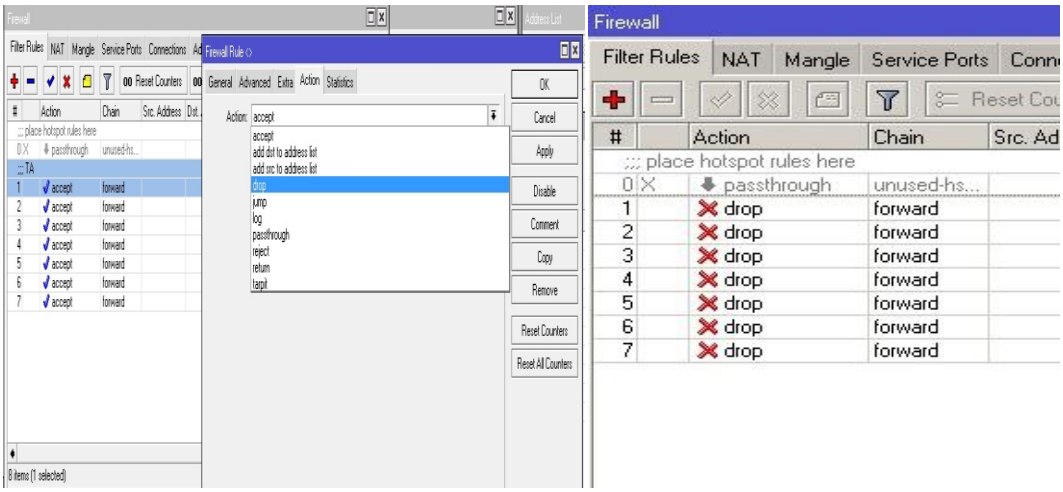
Pada tahapan ini hampir sama dengan menggunakan metode pengaturan bandwith simple queue namun pada tahap ini menggunakan menu pada queue tree. Untuk pembatasan bandwith sendiri untuk melakukan pengujian yaitu dengan memberikan bandwith sebesar 128 kbps, 256 kbps, 512 kbps, 1024 kbps dan 2048 kbps seperti simple queue. Pada queue tree hanya menggunakan satu buat konfigurasi yang dimana queue tree mengatur bandwith secara dynamic berbeda dengan simple queue yang harus mengatur secara static untuk setiap kelas pada iTCentrum yang ada.Gambar 4.26 merupakan tampilan gambar manajemen bandwith queue tree.



Gambar 4.26 Manajemen Bandwith Queue Tree

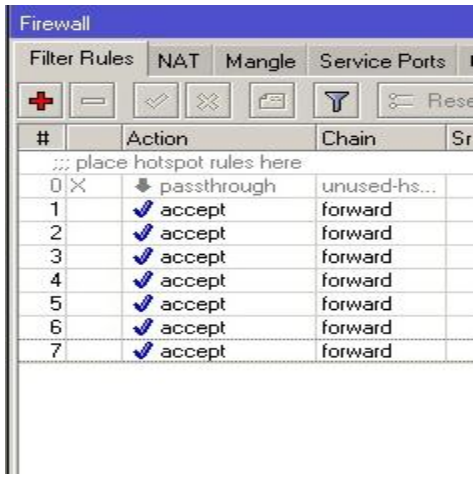
4.3.2.2 Firewall Filter Rule

Tahapan pengaturan Firewall ini tidak terdapat pada *simple queue.* Pengujian dilakukan dengan melakukan proses *drop* atu *accept* pada filter *rule*, filter *rule drop* untuk melakukan proses pembuangan paket data yang telah dideskripsikan yaitu seperti file ekstensi mp3, doc, exe, serta zip. Filter *Drop* dilakukan apabila sistem tidak ingin ada file-file ekstensi yang di deskripsikan masuk kedalam router. Gambar 4.27 merupakan tampilan untuk mengubah *action* paket yang diijinkan masuk maupun yang tidak dijinkan.



Gambar 4.27 Mengubah *Action*

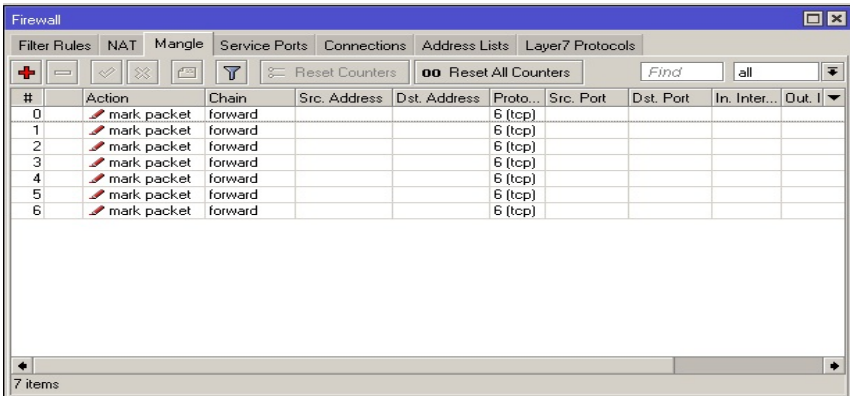
Gambar 4.28 untuk melakukan pengujian perhitungan kinerja Filter Rule *action* diganti menjadi *accept*, pada proses ini file ekstensi akan masuk kedalam proses *queue tree* karena file tersebut diperbolehkan untuk masuk kedalam router MikroTik.



Gambar 4.28 Accept Paket

4.3.2.3 Firewall *Mangle*

Firewall *Mangle* pada pengujian difungsikan sebagai rule lanjutan dari proses accept, pada Firewall Mangle dideskripsikan file ekstensi seperti mp3, doc, exe dan zip untuk ditangkap dan dilanjutkan ke dalam *queue tree*. Untuk proses *drop* tidak melalui proses mangle karena router akan langsung menolak paket tersebut tanpa pemeriksaan lanjutan. Pada *mangle* akan dideskripsikan nama file ekstensi yang nantinya akan diproses dalam *queue tree*. Mangle menyediakan juga mempunyai action untuk mendeskripsikan *layer 7 protocol* pada *marc packet* kolom *advanced*. Pada konfigurasi ini lah yang menangkap file ekstensi tersebut untuk dilanjutkan ke proeses *queue tree*. Gambar 4.29 merupakan aturan yang nantinya akan memproses paket data ke dalam queue tree.



Gambar 4.29 Firewall Mangle

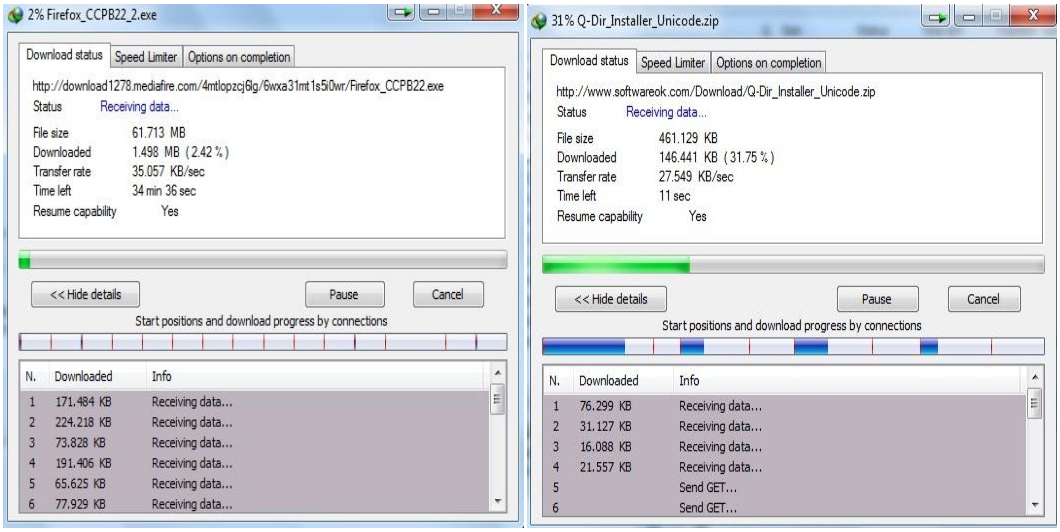
Tahapan-tahapan pengukuran kinerja setelahnya hampir sama pada skenario *simple queue* dengan FileZilla *Remote*, WireShark Analisis Paket Data, serta Ping Menggunakan Command Prompt. Pengujian sendiri hanya mengambil beberapa file ekstensi seperti mp3, doc, exe, serta exe yang akan dilanjutkan untuk pengukuran kinerjanya.

4.3.3 Hasil Pengujian Fungsionalitas

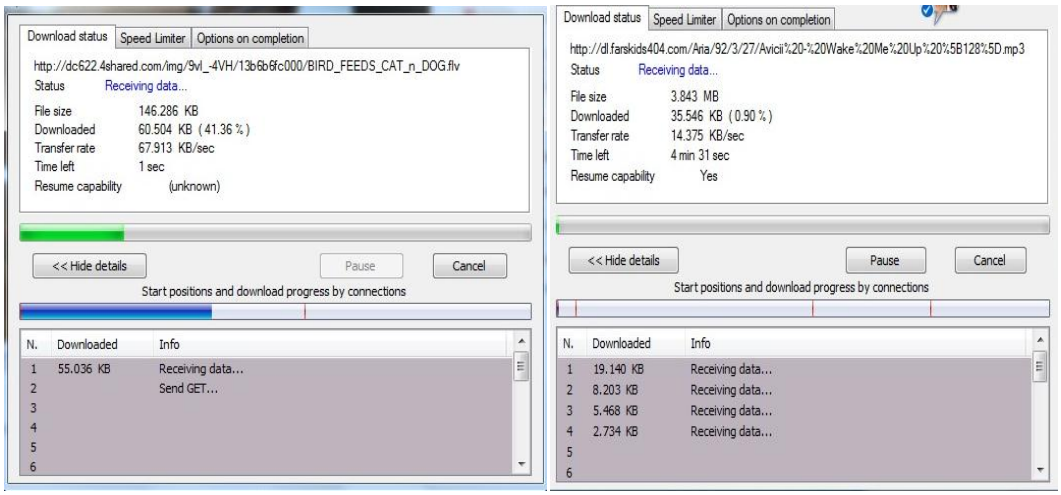
Seperti yang sudah dijelaskan pada bab sebelumnya pengujian fungsionalitas lebih ditujukan pada berhasil atau tidaknya sistem queue tree dengan Firewall *Protocol Layer 7*. Untuk hasil pengujian fungsionalitas terdapat dua hasil yaitu hasil paket yang dibuang dan hasil paket yang diterima.

4.3.3.1 *Action Accept*

Setelah konfigurasi dan pengujian implementasi dilakukan dengan menggunakan skenario yang telah dirancang maka akan didapatkan hasil pada gambar 4.30. Firewall Filter Rule *action accept* tentu sistem akan memperbolehkan paket data masuk ke dalam router dan user tentu dapat mengunduh data berdasarkan file ekstensinya. Pada hasil yang didapat saya mengambil sampel untuk file ekstensi .exe, .zip, .flv, dan .mp3 yang diijinkan untuk diunduh oleh komputer klien.



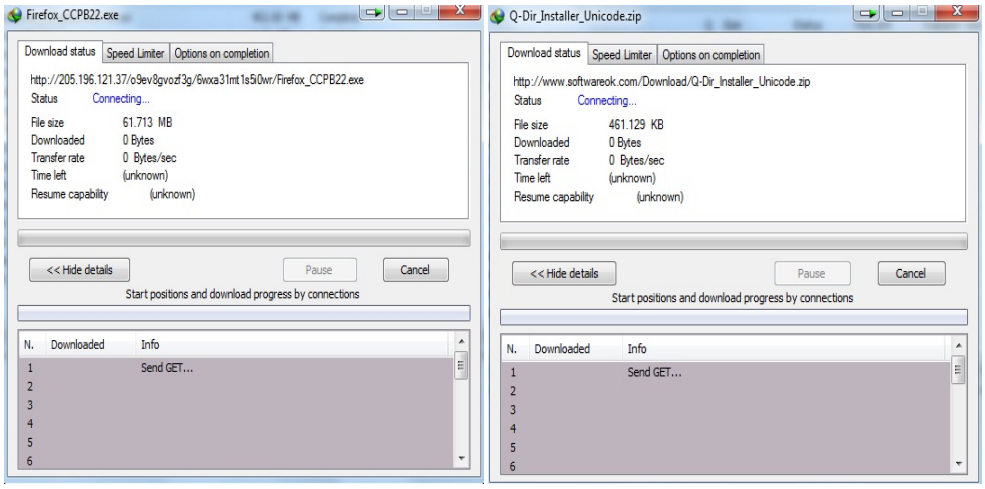
Gambar 4.30 *Action Accept* paket .exe dan .zip



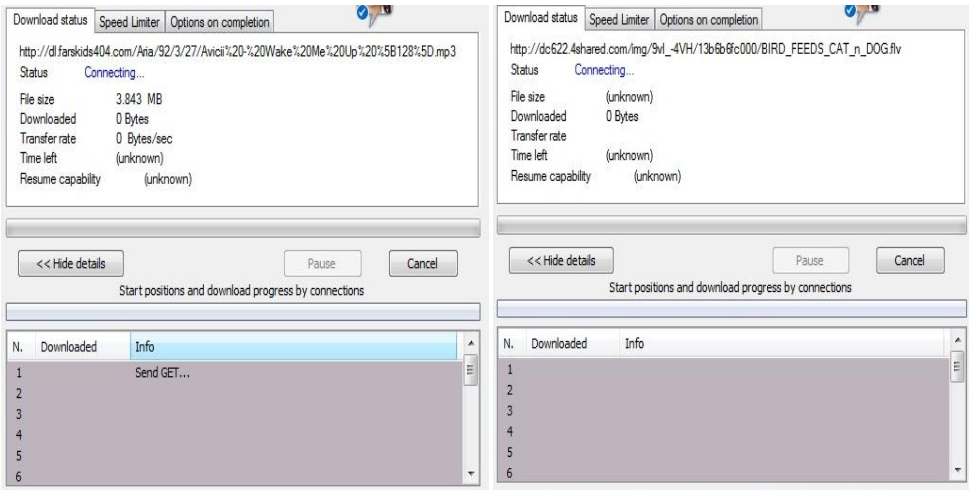
Gambar 4.31 *Action Accept* Paket .flv dan .mp3

4.3.3.2 *Action Drop*

Untuk konfigurasi action drop dari skenario yang telah dilakukan maka akan didapatkan hasil pada gambar 4.32 dimana file .exe, .zip, .flv, dan .mp3 sama sekali tidak mendapatkan ijin untuk diunduh oleh user. Terlihat pada gambar 4.32 tidak ada sama sekali bandwith yang masuk untuk melakukan proses pengunduhan tersebut.

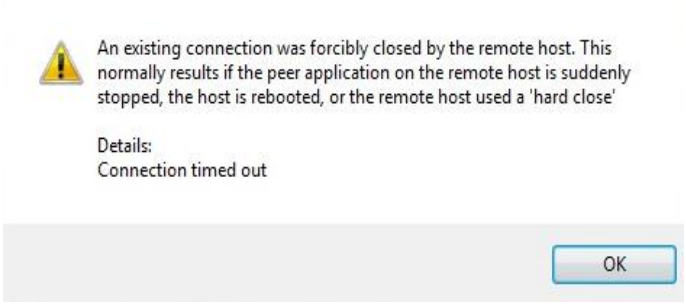


Gambar 4.32 *Action Drop* paket .exe dan .zip



Gambar 4.33 *Action Drop* Paket .flv dan .mp3

Gambar 4.34 memperlihatkan catatan bahwa tidak adanya koneksi untuk melakukan proses pengunduhan dan sistem akan menutup koneksi tersebut karena *header* tidak sesuai dengan aturan-aturan yang telah diterapkan untuk sistem yang ada di iTCentrum.



Gambar 4.34 *Connection Timed Out*

4.3.4 Hasil Pengujuan Kinerja

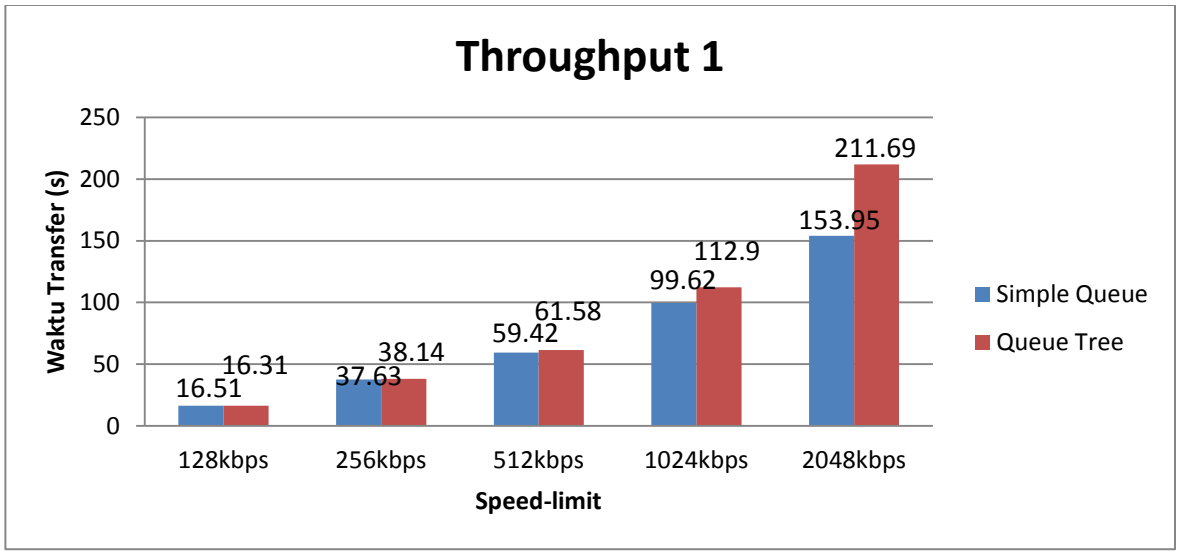
4.3.4.1 Throughput

Parameter pertama yang diamati berdasarkan pengujian implementasi masing-masing dari skenario *simple queue* dan *queue tree protocol layer 7* adalah *throughput*. Semakin besar nilai *throughput* maka semakin bagus kinerja suatu jaringan itu. *Throughput* didapat dari skenario perancangan untuk *simple queue* dan

queue tree protocol layer 7. Hasil kinerja throughput dari skenario tersebut dapat dilihat pada tabel 4.1 dan gambar 4.35.

Tabel 4.1 Hasil Implementasi *Throughput* 1

Manajemen Bandwith	Ukuran File (MB)	Percobaan	Limit Bandwith (kbps)	Waktu Transfer (s)	Throughput (kbps)
Simple Queue	3.387	1	128	205.2	16.51
			256	90	37.63
			512	57	59.42
			1024	34	99.62
			2048	22	153.95
Queue Tree			128	207.6	16.31
			256	88.8	38.14
			512	55	61.58
			1024	30	112.9
			2048	16	211.69



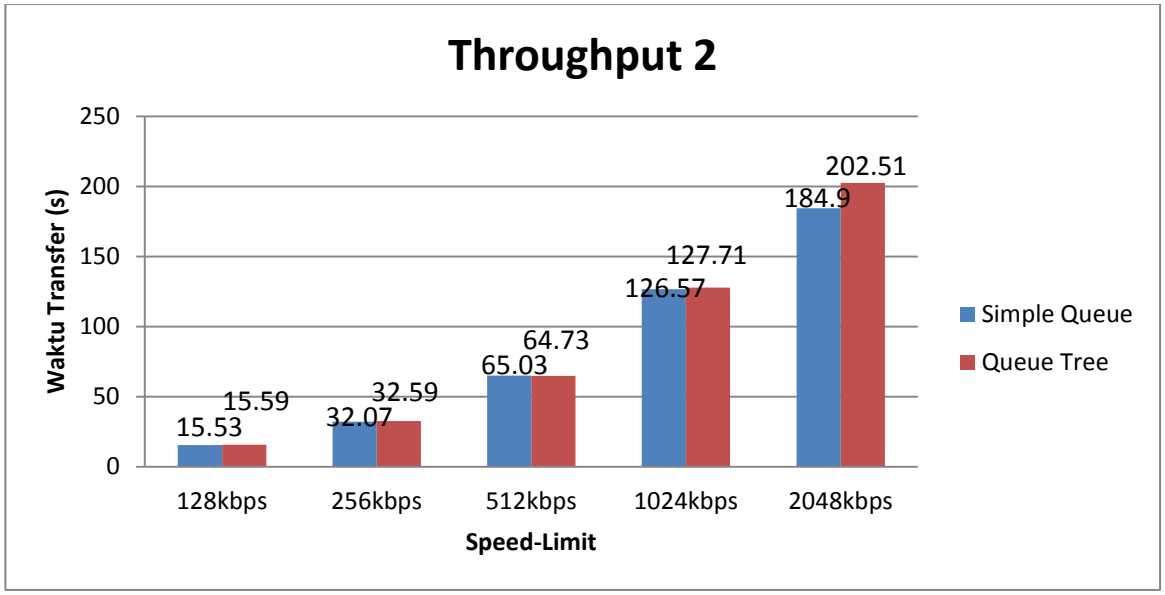
Gambar 4.35 Grafik *Throughput* 1

Berdasarkan data di atas, perbedaan Nilai *throughput* untuk percobaan 1 relatif tidak terlalu besar. Nilai *throughput* untuk Skenario 1 itu sendiri berkisar 16.00-212.00 kbps. Masing-masing pada limit bandwith 128 kbps *simple queue* dan *queue tree* adalah sebesar 16.51 kbps dan 16.31 kbps, terjadi peningkatan untuk

setiap limit bandwith 256 kbps *simple queue* dan *queue tree* yaitu sebesar 37.63 kbps dan 38.14 kbps, pada *simple queue* limit 512 kbps mendapatkan *throughput* sebesar 59.42 kbps dan pada queue tree mendapatkan *throughput* sebesar 61.58 kbps, pada *simple queue* dan *queue tree* limit bandwith 1024 kbps mendapatkan *throughput* sebesar 99.62 kbps dan 112.9 kbps, sedangkan peningkatan jauh lebih besar didapat dari *simple queue* dan *queue tree* limit bandwith 2048 kbps yang masing-masing mendapatkan nilai 153.95 kbps dan 211.68 kbps.

Tabel 4.2 Hasil Implementasi *Throughput* 2

Manajemen Bandwith	Ukuran File (MB)	Percobaan	Limit Bandwith (kbps)	Waktu Transfer (s)	Throughput (kbps)
Simple Queue	8.505	2	128	547.8	15.53
			256	265.2	32.07
			512	130.8	65.03
			1024	67.2	126.57
			2048	46	184.9
Queue Tree			128	545.4	15.59
			256	261	32.59
			512	131.4	64.73
			1024	66.6	127.71
			2048	42	202.51

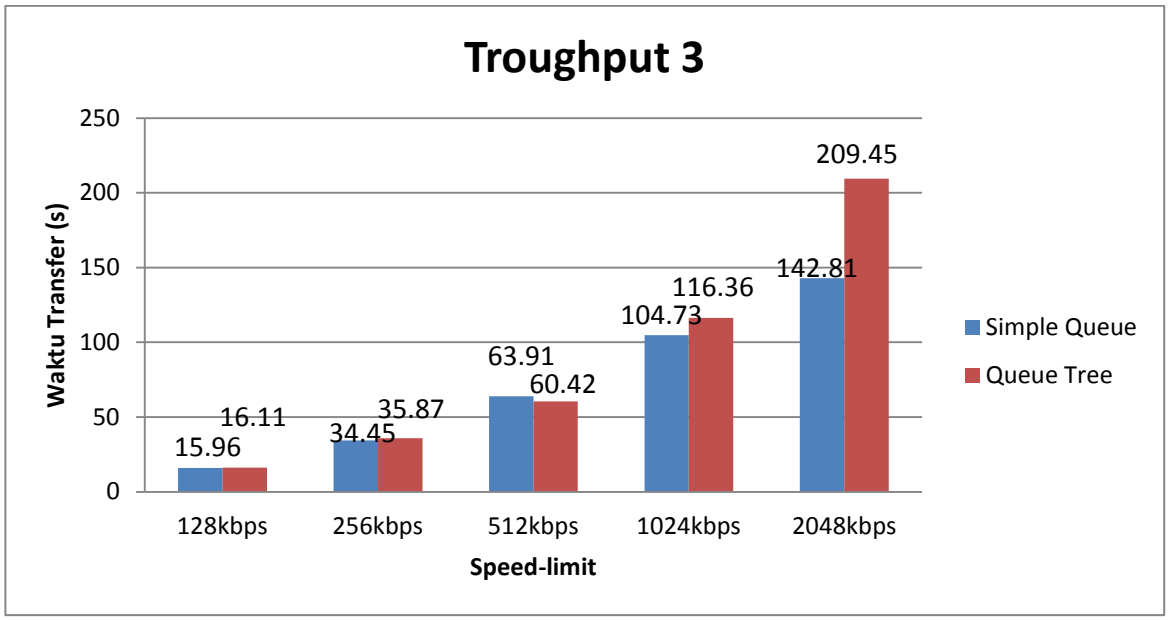


Gambar 4.36 Grafik *Throughput* 2

Perbedaan untuk *throughput* untuk percobaan 2 tidak terlalu besar. Nilai rata-rata *throughput* untuk Skenario 2 beriksar 15.00-203.00 kbps. *Simple queue* limit bandwith 128 kbps mendapatkan *throughput* sebesar 15.53 kbps pada *queue tree* mendapatkan *throughput* sebesar 15.59 kbps, pada limit bandwith 256 kbps manajemen *simple queue* mendapatkan *throughput* sebesar 32.07 kbps dan *queue tree* mendapatkan *throughput* sebesar 32.59 kbps, pada limit bandwith 512 kbps *simple queue* mendapatkan *throughput* sebesar 65.03 kbps pada *queue tree* sebesar 64.73 kbps, limit bandwith 1024 kbps *simple queue* mendapatkan *throughput* sebesar 126.57 kbps pada *queue tree* mendapatkan nilai sebesar 127.71 kbps, sedangkan untuk bandwith 2048 kbps *simple queue* dan *queue tree* mendapatkan bandwith sebesar 184.9 kbps dan 202.51 kbps.

Tabel 4.3 Hasil Implementasi *Throughput* 3

Manajemen Bandwith	Ukuran File (MB)	Percobaan	Limit Bandwith (kbps)	Waktu Transfer (s)	Throughput (kbps)
Simple Queue	3.142	3	128	196.8	15.96
			256	91.2	34.45
			512	53	63.91
			1024	30	104.73
			2048	22	142.81
Queue Tree			128	195	16.11
			256	87.6	35.87
			512	52	60.42
			1024	27	116.36
			2048	15	209.45



Gambar 4.37 Grafik *Throughput* 3

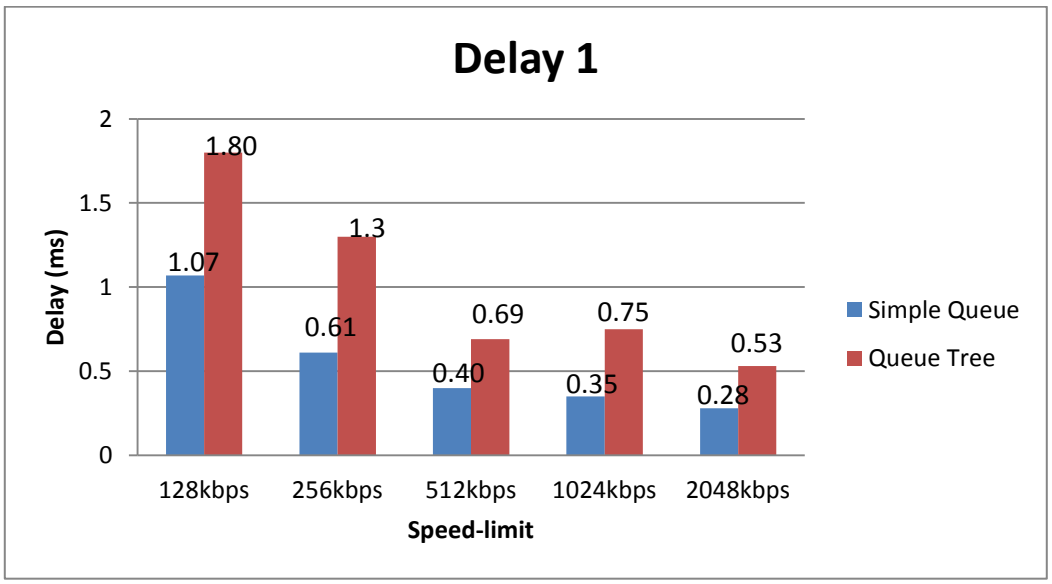
Untuk perbedaan nilai *throughput* percobaan 3 juga tidak terlalu besar. Nilai rata-rata untuk masing-masing limit bandwith 15.00-210.00 kbps. Skenario 3 dengan limit bandwith 128 kbps *simple queue* mendapatkan *throughput* sebesar 15.96 kbps pada *queue tree* mendapatkan 16.11 kbps, *simple queue* limit bandwith 256 kbps mendapatkan *throughput* sebesar 34.45 kbps pada *queue tree* mendapatkan *throughput* sebesar 35.87 kbps, pada *simple queue* limit bandwith 512 kbps mendapatkan nilai sebesar 63.91 kbps pada *queue tree* mendapatkan 60.42 kbps nilai *throughput*, *simple queue* limit bandwith 1024 kbps 104.73 kbps pada *queue tree* mendapatkan nilai *throughput* sebesar 116.36 kbps, sedangkan untuk limit bandwith 2048 kbps selalu terjadi peningkatan yang signifikan untuk *queue tree* sebesar 209.45 kbps dan *simple queue* nya mendapatkan nilai *throughput* sebesar 142.81 kbps.

4.3.4.2 *Delay*

Parameter kedua adalah *delay*, semakin kecil *delay* yang dihasilkan pada saat pengiriman data maka kinerja jaringan menjadi lebaih baik karena semakin cepat data sampai ke tujuan. Hasil pengambilan data dari pengujian implementasi kinerja menggunakan ping dan WireShark dapat dilihat pada tabel 4.4 dan gambar 4.38.

Tabel 4.4 Tabel Hasil Implementasi *Delay* 1

Manajemen Bandwith	Ukuran File (MB)	Percobaan	Limit Bandwith (kbps)	Delay (ms)
Simple Queue	1.840	1	128	1.07
			256	0.61
			512	0.40
			1024	0.35
			2048	0.28
Queue Tree			128	1.80
			256	1.30
			512	0.69
			1024	0.75
			2048	0.53



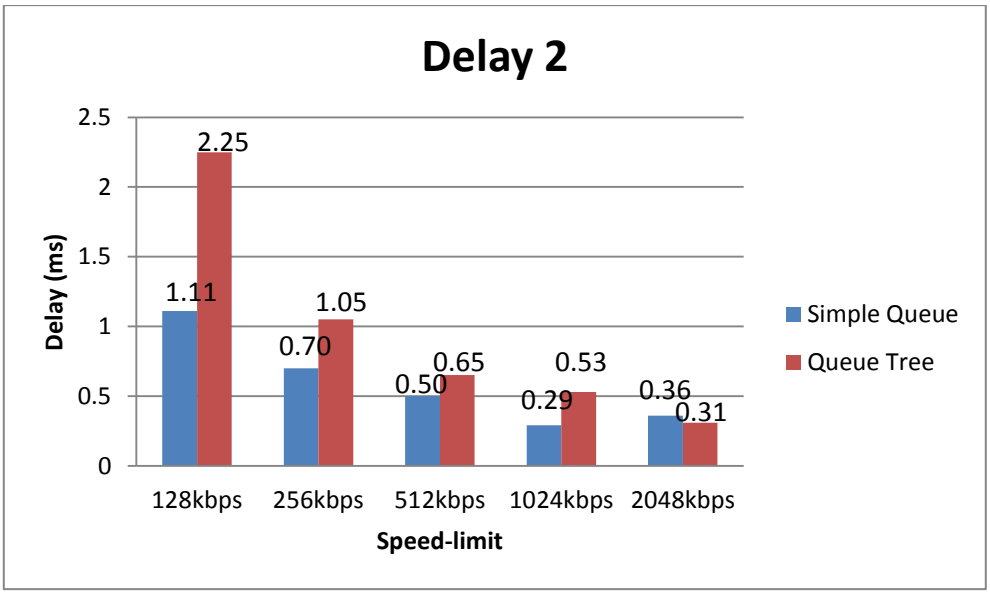
Gambar 4.38 Grafik *Delay* 1

Perbedaan *delay simple queue* dan *queue tree* percobaan 1 relatif kecil. Perbedaan tersebut berkisar antara 0.20 – 1.8 ms. *Simple queue* limit bandwith 128 kbps memiliki *delay* sebesar 1.07 ms dan *queue tree* memiliki *delay* sebesar 1.80, pada limit bandwith 256 kbps *simple queue* memiliki bandwith 0.06 ms dan *queue tree* memiliki *delay* sebesar 1.30 ms, pada limit bandwith 512 kbps *simple queue* memiliki *delay* sebesar 0.40 ms dan *queue tree* memiliki *delay* sebesar 0.69 ms, pada

limit bandwith 1024 kbps memiliki *delay* sebesar 0.35 ms dan *queue tree* memiliki delay sebesar 0.75 ms, pada limit bandwith 2048 kbps *simple queue* memiliki *delay* sebesar 0.28 ms dan *delay* sebesar 0.53 pada *queue tree*.

Tabel 4.5 Hasil Implementasi *Delay* 2

Manajemen Bandwith	Ukuran File (MB)	Percobaan	Limit Bandwith (kbps)	Delay (ms)
Simple Queue	1.309	2	128	1.11
			256	0.70
			512	0.50
			1024	0.29
			2048	0.36
Queue Tree			128	2.25
			256	1.05
			512	0.65
			1024	0.53
			2048	0.31



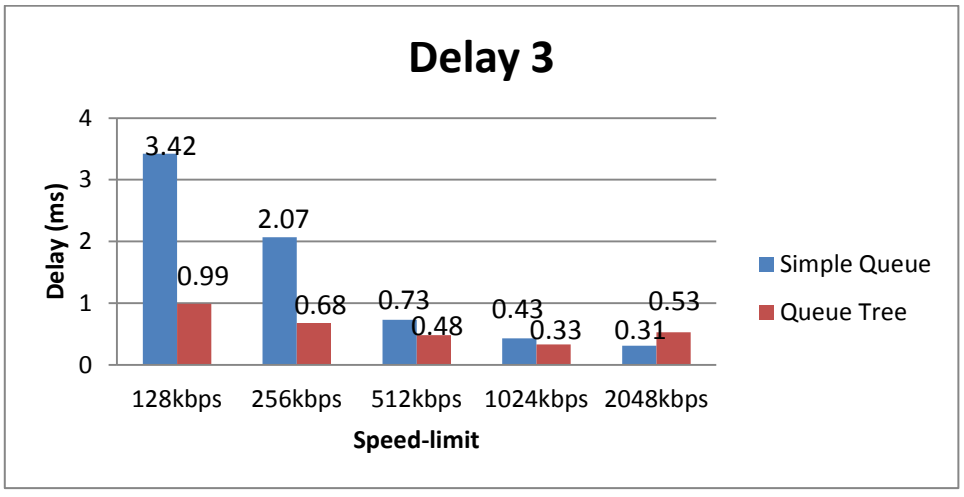
Gambar 4.39 Grafik *Delay* 2

Perbedaan *delay simple queue* dan *queue tree* percobaan 2 relatif besar. Perbedaan tersebut antara 0.5 – 2.25 ms. *Simple queue* limit bandwith 128 kbps memiliki *delay* sebesar 1.11 ms dan *queue tree* memiliki *delay* sebesar 2.25 ms, pada

limit bandwith 256 kbps *simple queue* memiliki bandwith 0.70 ms dan *queue tree* memiliki *delay* sebesar 1.05 ms, pada limit bandwith 512 kbps *simple queue* memiliki *delay* sebesar 0.50 ms dan *queue tree* memiliki *delay* sebesar 0.65 ms, pada limit bandwith 1024 kbps memiliki *delay* sebesar 0.29 ms dan *queue tree* memiliki *delay* sebesar 0.53 ms, pada limit bandwith 2048 kbps *simple queue* memiliki *delay* sebesar 0.36 ms dan *delay* sebesar 0.31 pada *queue tree*.

Tabel 4.6 Hasil Implementasi *Delay* 3

Manajemen Bandwith	Ukuran File (MB)	Percobaan	Limit Bandwith (kbps)	Delay (ms)
Simple Queue	1.269	3	128	3.42
			256	2.07
			512	0.73
			1024	0.43
			2048	0.31
Queue Tree			128	0.99
			256	0.68
			512	0.48
			1024	0.33
			2048	0.53



Gambar 4.40 Grafik *Delay* 3

Perbedaan *delay simple queue* dan *queue tree* percobaan 3 relatif sangat besar. Perbedaan tersebut antara 0.30 – 3.42 ms. *Simple queue* limit bandwith 128 kbps

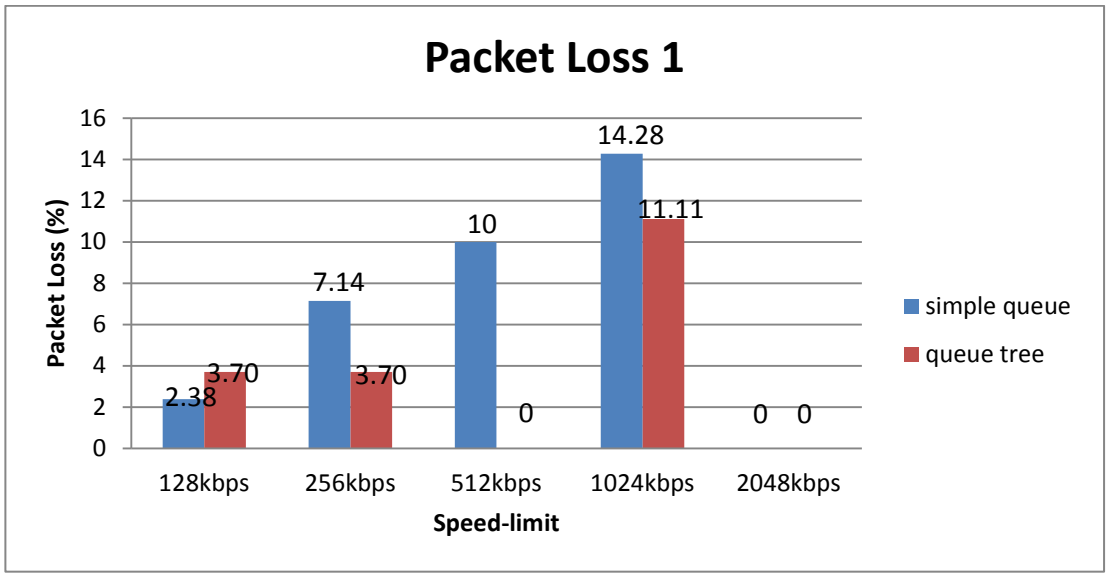
memiliki *delay* sebesar 3.42 ms dan *queue tree* memiliki *delay* sebesar 0.99 ms, pada limit bandwith 256 kbps *simple queue* memiliki bandwith 2.07 ms dan *queue tree* memiliki *delay* sebesar 0.68 ms, pada limit bandwith 512 kbps *simple queue* memiliki *delay* sebesar 0.73 ms dan *queue tree* memiliki *delay* sebesar 0.47 ms, pada limit bandwith 1024 kbps memiliki *delay* sebesar 0.42 ms dan *queue tree* memiliki *delay* sebesar 0.33 ms, pada limit bandwith 2048 kbps *simple queue* memiliki de*l*ay sebesar 0.30 ms dan *delay* sebesar 0.53ms pada *queue tree*.

4.3.4.3 *Packet Loss*

Parameter ketiga yang diamati adalah *packet loss*, semakin kecil *packet loss* yang terjadi pada saat pengirimian data maka kinerja jaringan menjadi lebih baik. Hal ini dikarenakan paket data yang hilang selama proses pengiriman menjadi lebih sedikit. Untuk mendapatkan hasil *packet loss* dilakukan pengiriman paket data sebesar 1.309 MB dipilih penulis untuk digunakan pada pengujian kinerja ini. Hasil pengambilan data parameter *packet loss* berdasarkan skenario *simple queue* dan *queue tree* yang diambil menggunakan ping pada command prompt dapat dilihat pada tabel dan gambar 4.41.

Tabel 4.7 Hasil Implementasi *Packet Loss* 1

Manajemen Bandwith	Ukuran File (MB)	Percobaan	Limit Bandwith (kbps)	Packet Loss (%)
Simple Queue	1.309	1	128	2.38
			256	7.14
			512	10
			1024	14.28
			2048	0
Queue Tree			128	3.70
			256	3.70
			512	0
			1024	11.11
			2048Kb	0



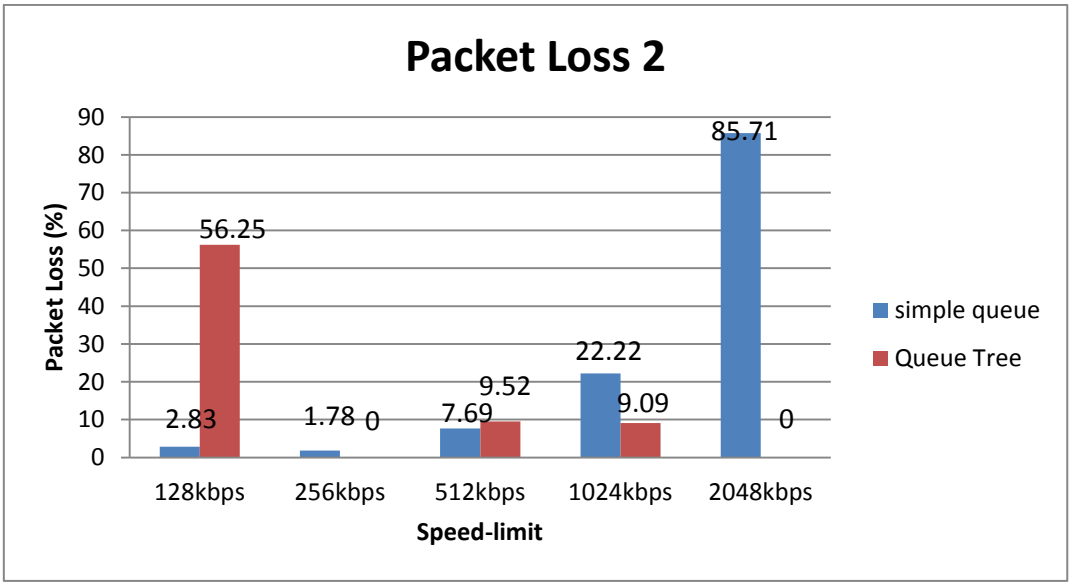
Gambar 4.41 Grafik *Packet Loss* 1

Hasil pengambilan data menunjukan nilai rata-rata *packet loss* untuk percobaan 1 yang dihasilkan untuk setiap limit bandwith ketika dilakukan percobaan dengan skenario manajemen bandwith *simple queue* dan *queue tree protocol layer 7*. Untuk perbedaan *packet loss simple queue* dan *queue tree* rata-rata berkisar antara 0% - 15%. *Simple queue* limit bandwith 128 kbps menerima packet loss 2,38% dan *queue tree* menerima *packet loss* sebesar 3.70%, *simple queue* limit bandwith 256 kbps menerima *packet loss* sebesar 7.14% dan pada *queue tree* menerima *packet loss* sebesar 3.70%, limit bandwith 512 kbps *simple queue* menerima *packet loss* sebesar 10% pada *queue tree* tidak menerima *packet loss* yaitu 0%, *simple queue* limit bandwith 1024 kbps menerima *packet loss* 14.28% dan *queue tree* menerima *packet loss* sebesar 11.11%, pada *simple queue* limit 2048 kbps dan *queue tree* sama-sama tidak menerima *packet loss* yaitu 0% kehilangan paket yang diterima.

Tabel 4.8 Hasil Implementasi *Packet Loss* 2

Manajemen Bandwith	Ukuran File (MB)	Percobaan	Limit Bandiwith (kbps)	Packet Loss (%)
Simple Queue	1.840	2	128	2.83
			256	1.78
			512	7.69

			1024	22.22
			2048	85.71
Queue Tree			128	56.25
			256	0
			512	9.52
			1024	9.09
			2048	0

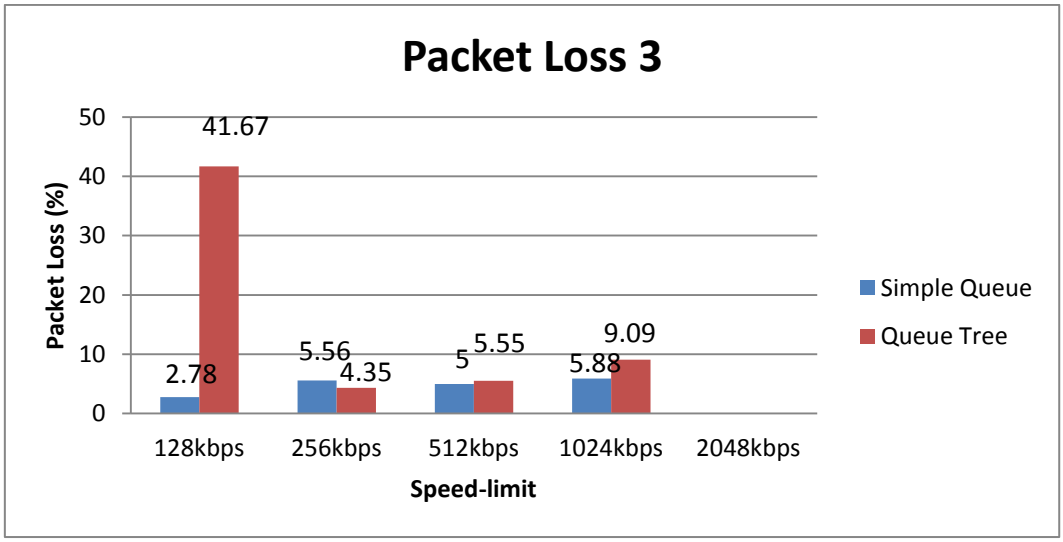


Gambar 4.42 Grafil *Packet Loss* 2

Rata-rata hasil pengambilan data *packet loss* percobaan 2 berkisar 0% - 86%. Hasil *packet loss* Skenario 2 untuk *simple queue* limit bandwith 128 kbps menerima 2.83% pada *queue tree* menerima *packet loss* sebesar 56.25%, *simple queue* limit bandwith 256 kbps menerima *packet loss* sebesar 1.78% pada *queue tree* tidak menerima 0% *packet loss*, *simple queue* limit bandwith 512 kbps menerima *packet loss* sebesar 7.69% pada queue tree menerima *packet loss* 9.52%, *simple queue* limit bandwith 1024 kbps menerima *packet loss* sebesar 22.22% dan pada *queue tree* 9.09% *packet loss*, *simple queue* limit bandwith 2048 kbps menerima *packet loss* 85.71% dan 0% *packet loss* yang diterima oleh *queue tree.*

Tabel 4.9 Hasil Implementasi *Packet Loss* 3

Manajemen Bandwith	Ukuran File (MB)	Percobaan	Limit Bandwith (kbps)	Packet Loss (%)
Simple Queue	1.269	3	128	2.78
			256	5.56
			512	5
			1024	5.88
			2048	0
Queue Tree			128	41.65
			256	4.35
			512	5.55
			1024	9.09
			2048	0



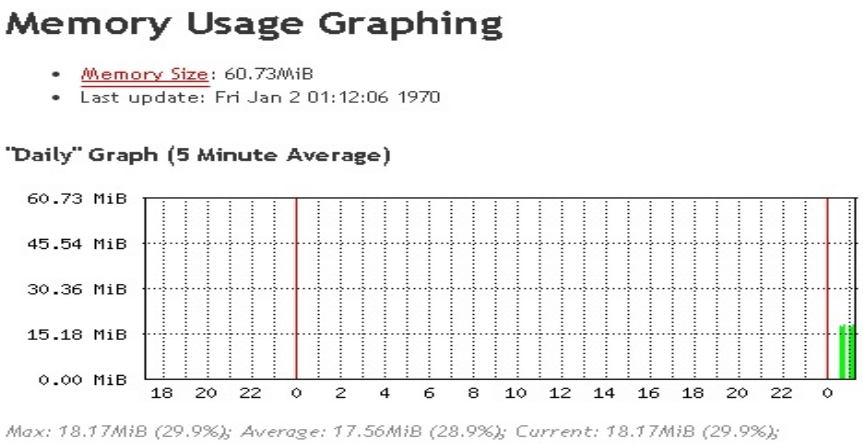
Gambar 4.43 Grafik *Packet Loss* 3

Untuk *packet loss* percobaan 3 rata-rata untuk masing-masing limit bandwith menerima *packet loss* antara 0% - 42%. *Simple queue* limit bandwith 128 kbps Koneksi Umum menerima *packet loss* 2.78% pada *queue tree* 41.67%, *simple queue* 256 kbps menerima *packet loss* sebesar 5.55% pada *queue tree* sebesar 4.35%, *packet loss limit* bandwith 512 kbps *simple queue* menerima 5% pada *queue tree* menerima

packet loss sebesar 5.55%, *simple queue* limit bandwith 1024 kbps menerima 5.88% pada *queue tree* menerima 9.09% *packet loss*, sedangkan untuk limit bandwith 2048 kbps masing-masing *simple queue* dan *queue tree* sama sekali tidak menerima *packet loss.*

4.3.4.4 *Resource Memory*

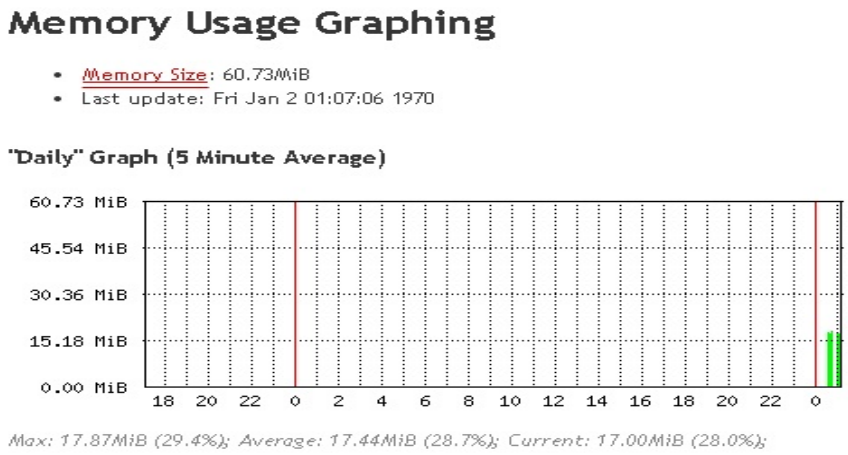
Parameter terakhir hasil pengujian implementasi dan pengambilan data dari skenario manajemen bandwith *simple queue* dan *queue tree protocol layer 7* adalah *resource memory router* yang menjalankan sistem *queue simple* ataupun *queue tree* dengan menggunakan tool graph yang tersedia pada MikroTikOS, pada tool graph ini memberikan rentang waktu selama 1 hari (*Daily*) dan akan melakukan update penggunaan memori setiap 5 menit sekali. Hasil yang didapat dari pengamatan *resource memory* yang dibebankan pada router saat menerapkan *simple queue* dan *queue tree* protocol layer 7 dapat dilihat pada gambar 4.44.



Gambar 4.44 *Resource Memori Simple Queue*

Pada gambar 4.38, terlihat penggunaan *resource memory* yang menerapkan skenario manajemen bandwith *simple queue.* Selama penggunaan, dari ukuran memory sebesar 60.73MiB (63.68) MB), terlihat (*Max*) pemakaian *resource memory* sebesar 18.17MiB (19.05 MB), rata-rata (*Avarege*) pemakaian sebesar 17.56MiB (18.41 MB), dan pemakaian (*Current*) sebesar 18.17Mib (19.05 MB).

Adapun untuk hasil kinerja penggunaan *resource memory* router yang menerapkan skenario manajemen bandwith *queue tree* protocol layer 7 dapat dilihat pada gambar 4.45.



Gambar 4.45 *Resource Memory Queue Tree*

Gambar 4.45 terlihat penggunaan *resource memory router* yang menerapkan *queue tree*, dari total *memory* yang ada sebesar 60.73MiB (63.68 MB), terlihat (*Max*) pemakaian *resource memory* sebesar 17.87MiB (18.74 MB), rata-rata (*average*) pemakaian sebesar 17.44MiB (18.29 MB) dan pemakaian (*Current*) sebesar 17.00MiB (17.83 MB).

4.4 Pembahasan

4.4.1 Firewall *Protocol Layer 7*

Peneliti melakukan penggunaan firewall *protocol layer 7 action accept* dan *action drop* pada *queue tree* dan hasilnya sistem mampu mengaplikasikan konfigurasi firewall secara baik. *Protocol layer 7 action accept* memeriksa pola *header* paket dari file ekstensi dilanjutkan dengan filter rule menggunakan *chain forward* dan selanjutnya firewall *mangle action accept* memproses paket file ekstensi tersebut yang masuk kedalam router, paket dengan *action accept* dapat diunduh oleh komputer user yang kemudian mendapatkan *speed-limit* pada *queue tree.*

Protocol layer 7 action drop memeriksa pola *header* paket dari file ekstensi dilanjutkan dengan filter rule chain forward dan firewall *mangle action drop* memproses paket file ekstensi yang masuk kedalam router, namun pemeriksaan paket hanya sampai disini saja. Hal ini dikarenakan firewall telah menolak paket untuk diterima router dan komputer user sehingga tidak akan sampai pada aturan *speed-limit* pada *queue tree*. Secara otomatis router akan memberikan notifikasi pada komputer user bahwa proses pengunduhan tidak dapat dilanjutkan karena tidak mendapatkan bandwith untuk proses pengunduhan tersebut.

4.4.2 *Throughput*

Tabel 4.10 *Throughput* Keseluruhan

Percobaan	Ukuran File (MB)	Limit bandwith (kbps)	Throughput (kilobit/s)	
			Simple Queue	Queue Tree
1	3.387	128	16.51	16.31
		256	37.63	38.14
		512	59.42	61.58
		1024	99.62	112.9
		2048	153.95	211.69
2	8.505	128	15.53	15.59
		256	32.07	35.59
		512	65.03	64.73
		1024	126.57	127.71
		2048	184.9	202.51
3	3.142	128	15.96	16.11
		256	34.45	35.87
		512	63.91	60.42
		1024	104.73	116.36
		2048	142.81	209.45

Berdasarkan data statistik di atas yang dihasilkan pada pengambilan data dengan menggunakan skenario manajemen bandwith *simple queue* dan *queue tree* firewall *protocol layer 7*, perbedaan nilai *throughput* rata-rata untuk percobaan 1 relatif tidak terlalu besar untuk masing-masing speed limit yang diberikan. Ukuran

file 3.387 MB Nilai peningkatan yang signifikan terjadi pada speed limit dari 1024 kbps ke 2048 kbps.

Rata-rata perbedaan nilai *throughput* untuk percobaan 2 tidak terlalu besar. Percobaan pengunduhan file berukuran 8.505 MB dan *speed-limit* yang diberikan nilai *throughput* yang terlihat pada percobaan 2 rata-rata hanya memiliki sedikit perbedaan. Peningkatan yang signifikan terjadi pada *speed-limit* 1024 kbps dan 2048 kbps. Sedangkan Nilai *throughput* untuk percobaan 3 juga tidak memiliki perbedaan yang cukup besar. Hasil dapat dilihat dengan melakukan pengunduhan file sebesar 3.142 MB yang dilakukan peneliti.

Analisis yang didapat dari hasil rata-rata *throughput* pada sistem, *queue tree* yang dirancang memiliki nilai *throughput* yang lebih baik dibandingkan *simple queue* walaupun banyak penambahan fitur firewall yang digunakan.

4.4.3 *Delay*

Tabel 4.11 *Delay* Keseluruhan

Percobaan	Ukuran File (MB)	Limit bandwith (kbps)	Delay (ms)	
			Simple Queue	Queue Tree
1	1.840	128	1.07	1.8
		256	0.61	1.3
		512	0.4	0.69
		1024	0.35	0.75
		2048	0.28	0.53
2	1.309	128	1.11	2.25
		256	0.7	1.05
		512	0.5	0.65
		1024	0.29	0.53
		2048	0.36	0.31
3	1.269	128	3.42	0.99
		256	2.07	0.68
		512	0.73	0.48
		1024	0.43	0.33
		2048	0.31	0.53

Berdasarkan hasil data yang didapat peneliti, rata-rata nilai *delay* pada percobaan masing-masing kelas tidak terlalu besar. *Delay* yang didapat dari hasil skenario *simple queue* dan *queue tree* firewall *protocol layer 7* menunjukan bahwa rata-rata nilai *delay* pada *simple queue* dengan *speed-limit* dari 128 kbps hingga 2048 kbps memperlihatkan bahwa *delay* pada *simple queue* lebih kecil dibandingkan dengan *delay* pada *queue tree*.

Pada percobaan 1, dengan mengunduh file sebesar 1.840 MB terjadi perbedaan *delay* sangat signifikan, *delay* pada *queue tree* lebih besar dibandingkan dengan nilai *delay* pada *simple queue*, pada percobaan 2 dengan mengunduh file 1.309 MB nilai *delay* pada *simple queue* lebih kecil dibandingkan dengan *queue tree*. Sedangkan, pada percobaan 3 nilai *delay* pada *queue tree* menunjukan *delay* yang lebih baik dibandingkan *simple queue*.

Anlisis menunjukan bahwa *delay* pengiriman data yang terjadi pada *simple queue* lebih cepat sampai ke tujuan dibandingkan dengan *queue tree*, hal ini dipengaruhi firewall pada *queue tree* terus melakukan pemeriksaan paket data terlebih dahulu sebelum masuk kedalam jaringan router MikroTik.

4.4.4 *Packet Loss*

Tabel 4.12 *Packet Loss* Keseluruhan

Percobaan	Ukuran File (MB)	Limit bandwith (kbps)	Packet Loss (%)	
			Simple Queue	Queue Tree
1	1.309	128	2.38	3.7
		256	7.14	3.7
		512	10	0
		1024	14.28	11.11
		2048	0	0
2	1.840	128	2.83	56.25
		256	1.78	0
		512	7.69	9.52
		1024	22.22	9.09
		2048	85.71	0
3	1.269	128	2.78	41.65
		256	5.56	4.35

		512	5	5.55
		1024	5.88	9.09
		2048	0	0

Hasil pengambilan data menggunakan skenario manajemen bandwith *simple queue* dan *queue tree* firewall *protocol layer 7* terjadi perbedaan yang signifikan terhadap *packet loss* yang terjadi ketika melakukan proses pengunduhan file pada percobaan 1, percobaan 2, percobaan 3. Percobaan dilakukan dengan melakukan proses pengunduhan data dengan file ekstensi yang berbeda-beda untuk setiap kelasnya. percobaan 1 melakukan pengunduhan data berukuran 1.309 MB, percobaan 2 1.840 MB, dan percobaan 3 1.269 MB. Rata-rata persentase *packet loss* menggunakan *simple queue lebih* besar dibandingkan dengan menggunakan *queue tree*. Sehingga dapat dikatakan bahwa mekanisme perlindungan data penggunaan *queue tree* untuk mencegah kehilangan data jauh baik dibandingkan dengan penggunaan *simple queue*.

4.4.5 Resource *Memory*

Tabel 4.13 *Resource Memory* Keseluruhan

Manajemen Bandwith	Max (MB)	Average (MB)	Current (MB)
Simple Queue	19.05	18.41	19.05
Queue Tree	18.74	18.29	17.83

Berdasarkan data untuk kedua skenario menggunakan manajemen bandwith *simple queue* dan *queue tree* firewall *protocol layer 7* bahwa *simple queue* dan *queue tree*, beban yang diberikan oleh skenario *simple queue* terhadap *memory* lebih besar dibandingkan skenario *queue tree*. *Simple queue* memberikan beban *Max* sebesar 19.05 MB, *Average* 18.41 MB, dan *Current* 19.05 MB sedangkan *queue tree* dengan penambahan firewall *protocol layer 7* memberikan beban *Max* 18.74 MB, *Average* 18.29 MB, dan *Current* sebesar 17.83 MB. Memory menggunakan skenario *simple queue* menjalankan lebih banyak konfigurasi pengaturan manajemen bandwith dibanding *queue tree* karena harus dilakukan secara *static*.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Setelah melakukan perancangan dan analisa antara manajemen bandwith simple queue dan queue tree dengan firewall protocol layer 7 yang diterapakan pada jaringan iTCentrum. maka ditemukan kesimpulan bahwa queue tree dengan protocol layer 7 bisa lebih menghemat bandwith internet yang ditunjukan dengan nilai throughput, packet loss, delay serta penggunaan beban memori yang diberikan pada router. Dengan menambahkan penggunaan firewall layer 7 protocol juga lebih mudah untuk melakukan pengamanan yang dimana admin dapat memberikan hak ases kepada file-file yang diinginkan maupun yang tidak diinginkan masuk kedalam jaringan iTCentrum.

Dengan mencoba beberapa skenario, penulis dapat memperoleh point-point kesimpulan sebagai berikut :

1. Pada kecepatan internet yang penulis ujikan dari kedua sistem manajemen bandwith pada saat downloading file, *queue tree* memberikan performa yang lebih bagus.
2. Manajemen bandwith menggunakan *Queue Tree* sudah tepat sesuai dengan implementasi *protocol layer 7* dan sifatnya membagi bandwith sesuai dengan jumlah klien yang aktif.
3. Queue Tree memiliki qualitas lebih baik rata-rata nilai *throughput*, sebesar 16.51-153.95 kbps untuk *throughput* 1, 15.53-184.9 kbps untuk *throughput* 2, dan 15.96-142.81 kbps untuk *throughput* 3.
4. Penggunaan memori untuk *simple queue* memberikan beban *Max* sebesar 19.05 MB, *Average* 18.41 MB, dan *Current* 19.05 MB sedangkan *queue tree* dengan penambahan firewall *protocol layer 7* memberikan beban *Max* 18.74 MB, *Average* 18.29 MB, dan *Current* sebesar 17.83 MB dimana queue tree lebih ringan dibandingkan dengan simple queue.

5. Untuk delay pada saat pengiriman data simple queue memiliki delay yang lebih baik disbandingkan dengan queue tree.
6. Untuk rata-rata Packet loss 1 sebesar 0-15%, packet loss 2 sebesar 0-86%, packet loss 3 sebesar 0-42%, yang menunjukan bahwa queue tree memiliki tingkat kehilangan paket lebih sedikit dibandingkan dengan simple queue.
7. Penggunaan firewall yag diterapkan penulis disini membuahkan hasil, dengan bantuan firewall dapat diimplementasikan untuk sistem yang dibangun, hanya saja sistem lebih efisien diterapkan pada penggunaan manajemen bandwith queue tree. Keamanan firewall protocol layer 7 sangat berguna untuk pembatasan file-file ekstensi yang sangat tidak diinginkan masuk kedalam jaringan iTCentrum.

5.2 Saran

Setelah melakukan precancangan dan analisa dari kedua sistem manajemen bandiwth simple queue dan queue tree dengan firewall protocol layer 7 pada jaringan iTCentrum, yang kemudian hasilnya dituangkan dalam laporan tugas akhir ini, maka penulis memberikan beberapa saran, yang mungkin bisa dijadikan masukan atau pertimbangan bagi para administrator jaringan iTCentrum baik dalam skala kecil maupun besar, dengan beberapa saran diantaranya :

1. Mengembangkan pengujian manajemen bandwith queue tree dengan firewall protocol layer pada iTCentrum lebih mengoptimalkan manajemen bandwith jaringan.
2. Menggunakan penggunaan firewall protocol layer 7 dapat membantu dalam memanajemen bandwith file-file yang diinginkan maupun yang tidak diinginkan masuk kedalam jaringan yang ada.
3. Mengoptimalkan penggunaan firewall yang ada pada mikrotik sangat membantu selain dalam pengamanan jaringan dari hal-hal yang tidak diinginkan maupun dari penggunaan bandwith yang disediakan dalam jumlah terbatas.

DAFTAR PUSTAKA

Anonim. 2013. Manual:IP. Available at http://wiki.mikrotik.com/wiki/Manual:IP/Firewall/L7 . Diakses 11 Maret 2013.

Anonim. 2013. Manual:IP. Available at http://wiki.mikrotik.com/wiki/Manual:IP/Firewall/Filter. Diakses 11 Maret 2013.

Anonim. 2013. Manual:IP. Available at http://wiki.mikrotik.com/wiki/Manual:IP/Firewall/Mangle. Diakses 11 Maret 2013.

Anonim. 2012. Tangkap Content Extention Via L7. Available at http://www.forummikrotik.com/general-networking/15225-%5Btrik%5D-tangkap content-extention-via-l7-limit-download-dgn%5Bl7%5D-connbyt-drop-idm-6-print.html. Diakses 10 Juni 2013.

Darmawan, Erristhya,. Purnama, Indra., Mahardika, Rohmat, Ihromi, Tomy., Wicaksana, S, Wayan, I. 2012. *Bandwith Manajemen Queue Tree vs Simple Queue*. Bekasi, Universitas Guna darma.

Dhiaheka. 2013. Regular Expression(RegEx). Available at http://www.dhiaheka.blogspot.com/2013/05regular-expressionregex-pada-c_2196. Diakses 30 Maret 2013.

Hamid. 2011. *L7 Filtering*.

Herlambang, Moch. Linto., L. Aziz, Catur. 2008. *Panduan Lengkap Menguasai Router Masa Depan Menggunakan MikroTikRouterOS™*. Yogyakarta: Penerbit C.V ANDI.

iTCentrum. 2012. *Modul Pelatihan MTCNA*. Yogyakarta, Universitas Islam Indoensia.

Laboratorium Sistem dan Jaringan Komputer. 2011. *Modul Praktikum Jaringan Komputer*. Yogyakarta, Universitas Islam Indonesia.

Sutedjo, Budi.,Oetomo, Dharma. 2006. *Konsep dan Aplikasi Pemrograman Client Server dan Sistem Terdistribusi*. Yogyakarta: Penerbit ANDI.

Syafrizal, Melwin. 2005. *Pengantar Jaringan Komputer*. Yogykarta: Penerbit ANDI.

Towidjojo, R. 2013. *Mikrotik KUNG FU Kitab 1*. Jasakom

Towodjojo, R. 2013. *Mikrotik KUNG FU Kitab 2*. Jasakom

LAMPIRAN

1. **CD**